AWR Design Environment
User Guide

Product Version 17

AWR Design Environment User Guide

Cadence Design Systems, Inc. (Cadence), 2655 Seely Ave., San Jose, CA 95134, USA.

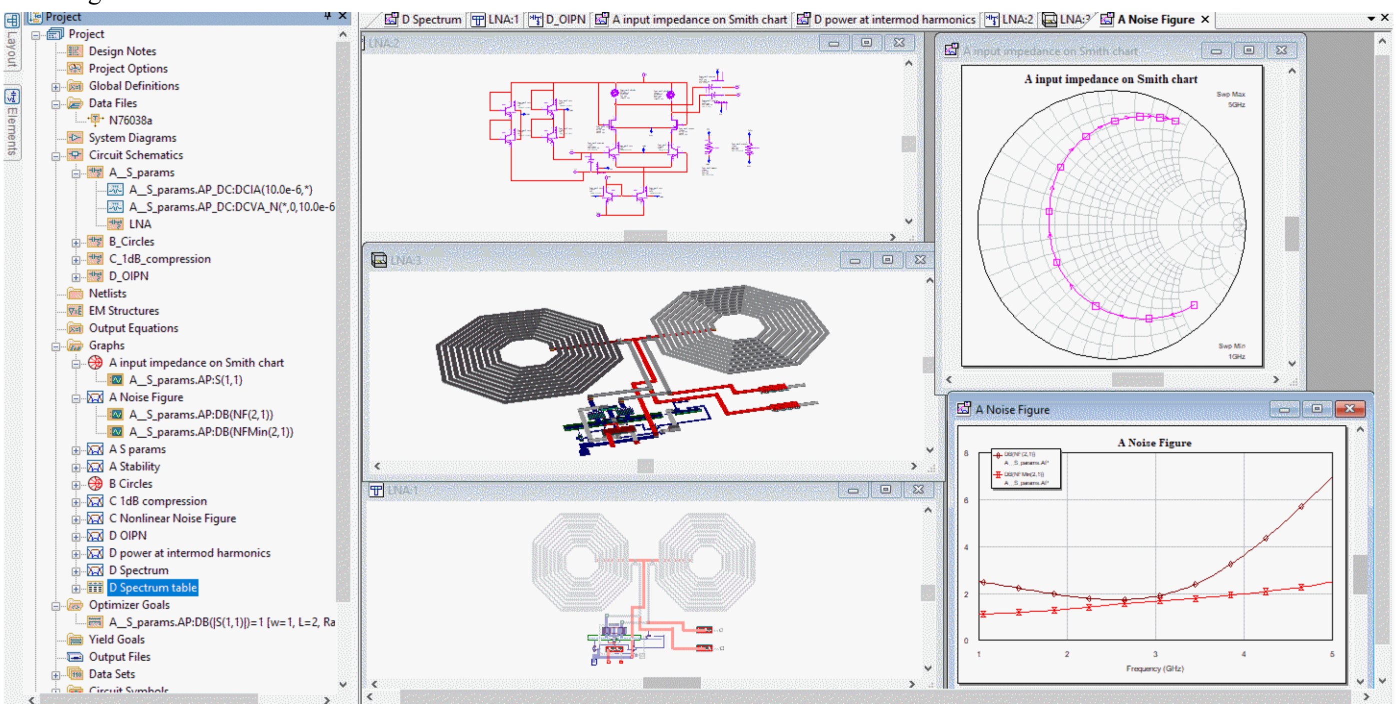
Table of Contents

Chapter 1. Preface

The Cadence® AWR Design Environment® platform incorporating Cadence Microwave Office® software and Cadence Visual System Simulator™ (VSS) communications and radar systems design software, is a powerful fully-integrated design and analysis tool for RF, microwave, millimeterwave, analog, and RFIC design that allows you to incorporate circuit designs into system designs without leaving the program.

Microwave Office software allows you to create complex circuit designs composed of linear, nonlinear, and EM structures, and generate layout representations of these designs. They allow you to perform fast and accurate analysis of your designs using linear, nonlinear harmonic balance, nonlinear Volterra-series, electromagnetic (EM), Cadence APLAC® HB, and Spectre simulation engines, and feature real-time tuning and optimizing capabilities.

The VSS software is the system level design component of the AWR Design Environment platform. With this software you can analyze a complete communications system, from data encoding through transmission, reception and data decoding.



1.1. About This Book

This book describes how to use the AWR Design Environment platform windows, menus, components, and scripts in preparation for performing linear, nonlinear, and EM design, layout, and simulation. It includes discussions of related concepts when appropriate.

Chapter 2 provides an overview of the program and describes how to set up and work with projects. Chapters 3 through 5 describe how to use the program to design circuits using data files, schematics and system diagrams, and netlists. Chapter 6 provides general information about EM analysis, and chapter 7 describes how to specify desired graphical or file-based output from the simulations that you perform. Chapters 8 and 9 provide information about working with data reports and annotations, respectively; and chapter 10 describes how to create and use your own circuit symbols. Chapter 11 covers both graph and simulation data sets, and chapter 12 describes how to use variables and equations to express parameter values in schematics and to post-process measurement data. Chapter 13 documents use of the Wizards available in the AWR Design Environment platform. Chapter 14 provides information about running developed scripts.

Appendix A presents the component libraries (XML and other) included as part of the AWR Design Environment platform. Appendix B is an overview of new design considerations that help designers create problem-free designs, appendix C provides information about AWR Design Environment platform warnings and errors, and appendix D includes an extensive collection of examples.

This guide assumes that you are familiar with Microsoft® Windows®, and have a working knowledge of high-frequency electronic design, layout, and analysis.

1.1.1. AWR Design Environment Limited Release

The current release includes some "Limited Release" features. In an effort to get customer feedback on features we are developing, and to ensure that those features are successfully solving the full range of the intended real-world engineering problems, Cadence is releasing select features in this "Limited Release" state. These features, while in the software, may require a license to access. To use these features please contact your local Cadence Sales representative to obtain documentation and the appropriate license(s). Cadence strongly encourages you to provide feedback to ensure that these features work well and solve your engineering problems.

1.1.2. Additional Documentation

The AWR Design Environment platform includes the following additional documentation:

- *What's New in AWR Design Environment v16?* presents the new features, user interface, elements, system blocks, and measurements for this release.
- The *AWR Design Environment Installation Guide* describes how to install the AWR Design Environment platform and configure it for locked or floating licensing options. It also provides licensing configuration troubleshooting tips. This document is downloadable from the Cadence AWR Knowledge Base.
- The *AWR Design Environment Getting Started Guide* familiarizes you with the AWR Design Environment platform through Microwave Office software, VSS software, Cadence Analyst™ 3D FEM EM analysis software, and Monolithic Microwave Integrated Circuit (MMIC) examples.

  Microwave Office example projects show how to design and analyze simple linear, nonlinear, and EM circuits, and how to create layouts. Visual System Simulator examples show how to design systems and perform simulations using predefined or customized transmitters and receivers. Analyst software examples show how to create and simulate 3D EM structures from Microwave Office, and MMIC examples show MMIC features and designs.

  You can perform simulations using a number of simulators, and then display the output in a wide variety of graphical forms based on your analysis needs. You can also tune or optimize the designs, and your changes are automatically and immediately reflected in the layout.
- The AWR Design Environment Simulation and Analysis Guide discusses simulation basics such as swept parameter analysis, tuning/optimizing/yield, and simulation filters; and provides simulation details for DC, linear, AC, harmonic balance, transient, and EM simulation/extraction theory and methods.
- The AWR Design Environment Dialog Box Reference provides a comprehensive reference of many AWR Design Environment platform dialog boxes with dialog box graphics, overviews, option details, and information on how to navigate to each dialog box.
- The AWR Microwave Office Layout Guide contains information on creating and viewing layouts for schematics and EM structures, including use of the Layout Manager, Layout Process File, artwork cell creation/editing/properties, Design Rule Checking, and other topics.

- The AWR Microwave Office Element Catalog provides complete reference information on the electrical element model database that you use to build schematics.
- The AWR Visual System Simulator System Block Catalog provides complete reference information on all of the system blocks that you use to build systems.
- The AWR Microwave Office Measurement Catalog provides complete reference information on the "measurements" (computed data such as gain, noise, power, or voltage) that you can choose as output for your simulations.
- The AWR Visual System Simulator Measurement Catalog provides complete reference information on the measurements you can choose as output for your simulations.
- The AWR Visual System Simulator Modeling Guide contains information on simulation basics, RF modeling capabilities, and noise modeling.
- The AWR API Scripting Guide explains the basic concepts of AWR Design Environment platform scripting and provides coding examples. It also provides information on the most useful objects, properties, and methods for creating scripts in the AWR Script Development Environment (AWR SDE). In addition, this guide contains the AWR Design Environment platform Component API list.
- The *AWR Design Environment Quick Reference* document lists keyboard shortcuts, mouse operations, and tips and tricks to optimize your use of the AWR Design Environment platform. This document is available within the program by choosing **Help > Quick Reference**.

1.1.3. Typographical Conventions

This guide uses the following typographical conventions.

Item	Convention
Anything that you select (or click on) in the AWR Design Environment platform, such as menus, nested submenus, menu options, dialog box options, buttons, and tab names	Shown in a bold alternate type. Nested menu selections are shown with a "**>**" to indicate that you select the first menu item and then select the submenu item: Choose **File > New Project**.
Text that you enter using the keyboard	Shown in bold type within quotation marks: Enter "**my_project**" in **Project Name**.
Keys or key combinations that you press	Shown in a bold alternate font with initial capitals. Key combinations using a "+" indicate that you press and hold the first key while pressing the second key: Press **Alt+F1**.
File names and directory paths	Shown in italics: See the *DEFAULTS.LPF* file.
Contents of a file, fields within a file, command names, command switches/arguments, or output from a command at the command prompt	Shown in a mono-spaced font: Define this parameter in the `$DEFAULT_VALUES` field.

1.2. Getting Online Help

The AWR Design Environment platform online Help provides information on the windows, menus, and dialog boxes that compose the design environment, as well as on the concepts involved.

To access online Help, choose **Help** from the menu bar, or press **F1**. The **Help** menu includes the following options:

Menu Choice	Description
Contents and Index	Open the online Help file to view AWR Design Environment platform Help topics organized by book/subject, or to search the Help contents by index or character string.
Help on Selected Item	Access Help on the currently selected item.
Getting Started	View the "Getting Started Guide" that includes introductory material for all products.
What's New	View the "What's New" document for information about new or enhanced features, elements, and measurements in the latest release.
Quick Reference	View the *AWR Design Environment Quick Reference* document to learn keyboard shortcuts, mouse operations, and tips and tricks to optimize your use of the AWR Design Environment platform.
Home Page	Open the Cadence AWR web page in your internet browser to the AWR® software products web page.
AWR e-Learning	Open the e-Learning portal login page in your internet browser to access AWR software training videos.
Knowledge Base	Open the AWR Knowledge Base in your internet browser to access to a variety of AWR software documentation and other resources.
Get Technical Support	Access the Cadence Learning and Support portal to search for answers to software or licensing issues or file a Support case.
Check for Update	Check for Cadence AWR software updates using an Autoupdate utility.
Show Files/Directories	Display/open a list of the files and directories the program uses.
Open Example	Display the Open Example Project dialog box to locate a specific project in the */Examples* subdirectory by filtering by keywords or project name.
Show License Agreement	Display the AWR Design Environment platform End User License Agreement.
About	Display AWR software copyright, release, license, and registration information.

In addition, the following context-sensitive Help is available:

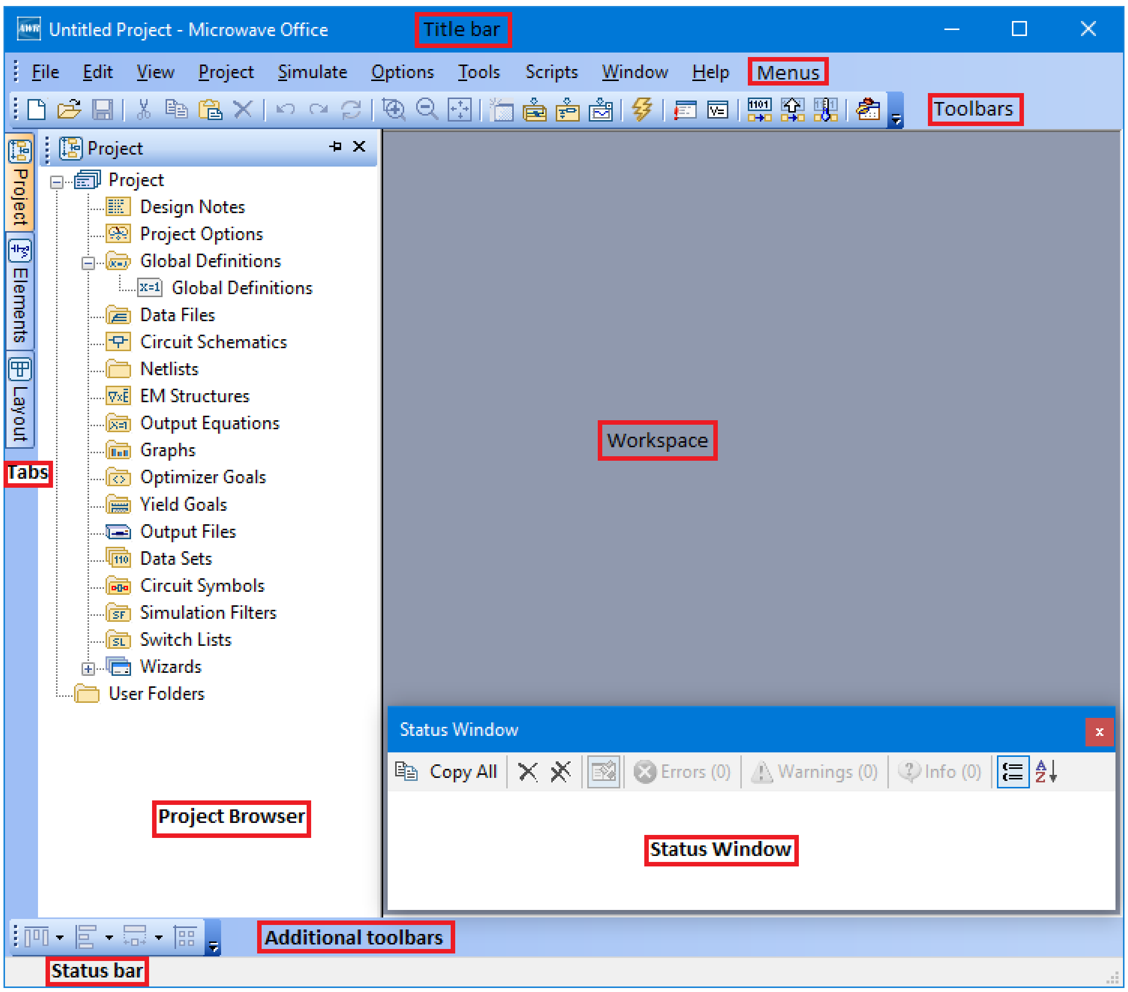
- Context-sensitive **Help** buttons in each dialog box. For example, to view Help for a specific measurement, select the measurement in the Add/Modify Measurement dialog box, and click the **Meas. Help** button.
- Context-sensitive Help for each element or system block in the Element Browser, accessed by right-clicking an element and choosing **Element Help**. You can also access element Help by choosing **Help > Element Help** after creating a schematic or system diagram, or by clicking **Element Help** in the Element Options dialog box.
- Context-sensitive Help for using the AWR Script Development Environment, accessed by selecting a keyword (for example; object, object model, or Visual Basic syntax), and pressing **F1**.

Chapter 2. The Design Environment

The Cadence® AWR Design Environment® platform is comprised of two powerful tools that can be used together to create an integrated system and RF design environment: Cadence Visual System Simulator™ (VSS) communications and radar systems design software, and Cadence Microwave Office® software. These powerful tools are fully integrated in the AWR Design Environment platform and allow you to incorporate circuit designs into system designs without leaving the design environment.

2.1. Components of the Design Environment

When you start the AWR Design Environment software, the main window shown in the following figure displays. In this window you build linear and nonlinear schematics, EM structures, system diagrams, generate layouts, perform simulations, display graphs, and optimize your designs. For each object you add to your project, a separate window opens in the program workspace.



The major components of the design environment are described in the following table:

Component	Description
menu bar	A set of menus located along the top of the window that allow you to perform all of the commands that drive the various AWR Design Environment platform tasks. Menus display based on the active window. Many of the menu choices and commands available from the menus are also available via the toolbar and/or in the Project Browser.
toolbar	A row of buttons you can dock on any edge of the workspace or float anywhere in the workspace that provides shortcuts to frequently used commands such as creating new schematics, performing simulations, or tuning parameter values or variables. To display or hide toolbar button categories, right-click the toolbar and select/deselect the toolbar category (for example, Standard, Equations, Schematic Design, or Graphs). To view a description of a toolbar command, move the mouse over the button and a pop-up description displays.
Project Browser	Located by default at the left of the workspace, this window comprises the complete collection of data and components that define the currently active project. Items are organized into a tree-like structure of nodes and include schematics, system diagrams and EM structures, simulation frequency settings, output graphs, user folders and more. The Project Browser is active when the program first opens, or when you click the **Project** tab in the workspace window. Right-click a node in the Project Browser to access menus of relevant commands. For more information about AWR Design Environment platform projects and the collection of nodes in the Project Browser, see "Working With Projects".
workspace	The area in which you design schematics, draw EM structures, view and edit layouts, and view graphs in individual windows.
Status Window	Located by default at the bottom of the workspace, this window displays information, errors and warnings from operations and simulations in the program. See "Viewing the Status Window" for more information.
tabs	You can dock or float the Project Browser, Elements Browser, Layout Manager, and Status Window. When docked, these windows can be placed into auto-hide mode by pressing the push pin icon in the upper right corner of the window. When in auto-hide mode, the windows disappear shortly after losing focus (clicking elsewhere) and become tabs along the edge of the main window. To display a hidden window, click on or hover the mouse cursor over the tab. You can also open hidden windows by choosing the associated option in the **View** menu. Click the **Project** tab to access the Project Browser, previously described. Click the **Elements** tab to access the Elements Browser, a comprehensive inventory of electrical entities for building schematics and system diagrams. For more information on the Elements Browser, see "Adding Elements Using the Elements Browser" and the AWR Microwave Office Element Catalog for Microwave Office elements and "Adding System Blocks Using the Elements Browser" and the AWR Visual System Simulator System Block Catalog for VSS system blocks. Click the **Layout** tab in the Microwave Office program to specify options for viewing and drawing layout representations and to create new layout cells. For more information about the Layout Manager, see *"Layout Overview "*.
Status bar	The bar along the very bottom of the design environment window that displays information dependent on what object is selected or what command is being executed. For example, when an element in a schematic is selected, the element name and ID display. When a polygon is selected, layer and size information displays, and when a trace on a graph is selected, the value of any swept parameter displays. While executing a command, the status bar displays hints on how to interact with the command. For example, while placing an element on a schematic, the hint text tells you how to rotate or flip the element.

2.1.1. Licensing and Version Information

Choose **Help > About** to display the AWR Design Environment platform version you are running. This dialog box also displays the AWR® software products features you are using and lists the date your current license file expires.

Choose **File > License > Configuration** to display the AWR License Configuration dialog box and view your computer's HostID, the location of your license file (for locked or floating licenses), and a detailed report of your license status. See "AWR License Configuration Dialog Box" for more details.

Choose **File > License > Feature Setup** to display the Select License Features dialog box. This dialog box helps you determine which license features you want to run and how you want to use them at software start-up. See "Select License Features Dialog Box" for more details.

2.2. Working With Projects

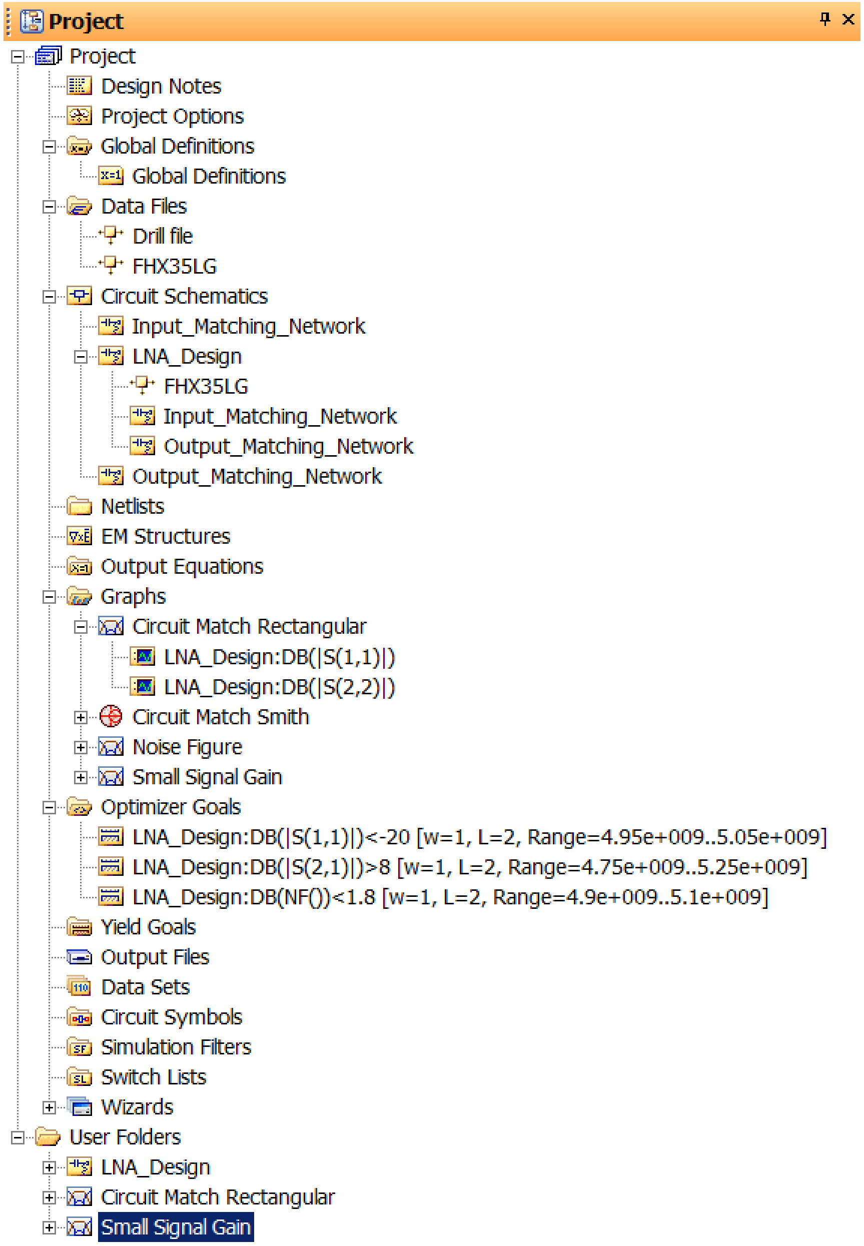
In the AWR Design Environment platform, you use projects to organize and manage related designs in a tree-like structure.

A project encompasses any desired set of designs and can include one or more schematics, netlists, EM structures, data files, or system diagrams. A project also includes anything associated with the designs, such as imported files, layout views, graphs, output files, and data sets. When you save a project, everything associated with it is automatically saved as well. AWR Design Environment platform projects are saved as **.emp* files.

After you create a project, you can create your designs. In the Microwave Office design suite you can generate layout representations of these designs, and output the layout to a DXF, GDSII, Gerber, or PADs file. You can perform simulations to analyze the designs and see the results on a variety of graphical forms that you specify. Then, you can tune or optimize parameter values and variables as needed to achieve the response you want. Since all parts of the Microwave Office program are fully integrated, your modifications are automatically reflected in both the schematic and the layout representation.

2.2.1. Using the Project Browser

The Project Browser (located on the left side of the main window when docked) is active when you click or hover the cursor over the **Project** tab along the edge of the main window. The Project Browser is always active when the program starts, and contains the entire collection of data that defines the current project, including schematics, system diagrams, EM structures, graphs, and others. This data is organized in a tree-like structure of items, as shown in the following figure.



2.2.1.1. Project Browser Contents

The Project Browser contains the following nodes:

Item	Description
Design Notes	Displays a rich text editor in which you can make design-related notes.
Project Options	Allows you to specify default frequencies used for project simulations, default schematic/diagram display options, default global units, interpolation/passivity defaults, and yield options.

Item	Description
Global Definitions	Allows you to define, import, embed, link, and export global variables and/or functions to be used as parameter values in schematics created within a project. You can also add substrate materials to this node and reference them from any schematic. For more information, see *“Variables and Equations”*.
Data Files	Allows you to import data files for use as subcircuits in schematics (typically S-parameter data) or for use in equations to retrieve row or column data from the file. The imported data files display as subnodes under **Data Files**. Data files imported for use as subcircuits can be Touchstone or MDIF (classical and generalized) format or raw data files. These files can also be directly used as the data source of a measurement. For example, you may import a two-port Touchstone file and create an S(2,1) measurement that uses it without first instantiating the data file as a subcircuit in a schematic. Also allows you to import data files to be used for performance comparison purposes. Data files imported for comparison purposes can be DC-IV format, text data or raw data files. (DC-IV is a Microwave Office format for reading DC-IV curves that measure a transistor or diode.) For more information, see “Importing Data Files”.
System Diagrams	Allows you to create system diagrams within a project. These diagrams display as subnodes under **System Diagrams**. For more information, see *“Schematics and System Diagrams”*.
Circuit Schematics	Allows you to create circuit schematics and netlists within a project. These schematics and netlists display as subnodes under **Circuit Schematics**. For more information, see *“Schematics and System Diagrams”* and *“Netlists”*.
Netlists	Allows you to create netlists within a project. These netlists display as subnodes under **Netlists**. For more information, see *“Netlists”*.
EM Structures	Allows you to create EM structures within a project. These structures display as subnodes under **EM Structures**. For more information, see “Creating EM Structures without Extraction”.
Output Equations	Allows you to specify equations used to post-process measurement data prior to displaying it in tabular or graphical form. For more information, see “Using Output Equations”.
Graphs	Allows you to create graphs to display the output of simulations performed within a project. Graphs display as subnodes under **Graphs**. You can create the following graph types: rectangular, Smith Chart, polar grid, histogram, antenna plot, tabular, constellation, and 3D. For more information, see *“Graphs, Measurements, and Output Files”*.
Optimizer Goals	Allows you to specify optimization goals for a project. The goals display as subnodes under **Optimizer Goals**. For more information, see “Optimization”.
Yield Goals	Allows you to specify yield goals for a project. The goals display as subnodes under **Yield Goals**. For more information, see “Yield Analysis”.
Output Files	Allows you to specify output files to contain the output of simulations performed within a project, as an alternative to graphical output. The output files display as subnodes of **Output Files**. Output files can be Touchstone format (S, Y, or Z-parameters, for circuit and EM simulations), SPICE Extraction files (for EM simulations), AM to AM, AM to PM, or AM to AM/PM files (for nonlinear circuit simulations), spectrum data files (for nonlinear circuit simulations), or antenna pattern files (for EM simulations). For more information, see “Working with Output Files ”.
Data Sets	Allows you to view and edit data sets in the project. Data sets are saved simulation results. See the *“Data Sets”* chapter for more information.

Item	Description
Circuit Symbols	Allows you to view, edit, and create custom circuit element and system block symbols that are stored in the project. See the *"Circuit Symbols"* chapter for more information.
Simulation Filters	Allows you to view, edit, and create simulation filters. Simulation filters give you control over what types of simulations are performed when you choose **Simulate > Analyze**. See "Simulation Filters" for more information.
Switch Lists	Allows you to view, edit, and create Switch Lists. A Switch List is a named list of switch views that a measurement can use to dynamically alter the schematic hierarchy. See "Switch View Concepts " for more information.
Wizards	Allows you to run AWR- or externally-authored wizards that add advanced functionality to the AWR Design Environment platform. The wizards display as subnodes under **Wizards**. For more information, see *"Wizards"*.
User Folders	Allows you to create your own folder structure. At any folder level, you can add any of the previously listed objects to custom organize your folders. For more information, see "User Folders".

2.2.1.2. Expanding and Collapsing Nodes

To expand a node in the Project Browser, do one of the following:

- **Shift**-right-click the node, and choose **Expand All**, or
- Click the **+** symbol to the left of the node.

To collapse a node in the Project Browser, do one of the following:

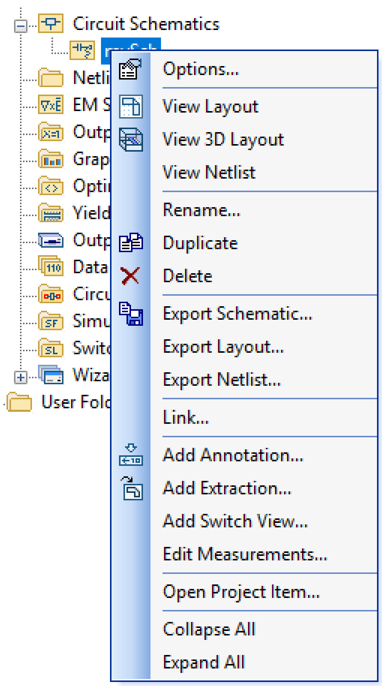
- **Shift**-right-click the node, and choose **Collapse All**, or
- Click the **-** symbol to the left of the node.

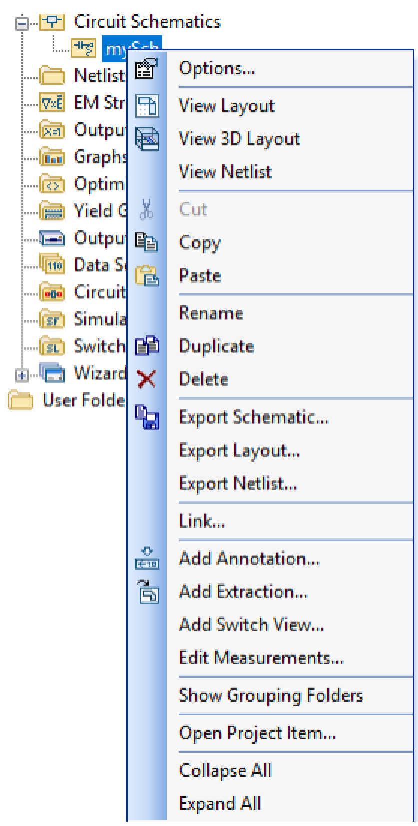
2.2.1.3. Speed Menus

To access speed menus from Project Browser nodes, simply right-click the node. You can access the most commonly used commands from speed menus, such as **Options** (properties), **Rename**, or **Delete**.

Not all commands are shown on the default speed menu. To access the full list of commands, **Shift**-right-click the node to view a full list.

The following figure shows an example of the difference between the simplified speed menu and the full speed menu for schematics.





2.2.1.4. Copying Project Items

To copy project items such as schematics, system diagrams, netlists, EM structures, text data files, and others, select the item node in the Project Browser and drag and drop it on the target project node. For example, to copy a schematic, drag the individual schematic node to the **Circuit Schematics** node in the Project Browser. A subnode named "schematicname_1" is created for the first copy. The object name is incremented by one (_2, _3 and so on) for each additional copy. After the new item is created, the name is directly editable.

You can also copy project items by right-clicking the item in the Project Browser and choosing **Duplicate** (the default hotkey is **Ctrl+D**). The naming operation is identical to the drag and drop copy method.

Measurements are not copied in this manner as you do not control a measurement name. See "Copying Measurements" for more details.

2.2.1.5. Renaming Project Items

To rename project items such as schematics, system diagrams, netlists, EM structures, text data files and others, right-click the item in the Project Browser and choose **Rename** , or press the **F2** key. A Rename dialog box displays for entering the new name and includes a 'Synchronize' option (if applicable) that propagates the name change throughout the project. If you press **Shift+F2** the item name is directly editable in the Project Browser without prompting from the Rename dialog box ('Synchronize' defaults to selected in this mode).

2.2.1.6. Deleting Project Items

To delete project items such as schematics, system diagrams, netlists, EM structures, text data files, and others, right-click the item in the Project Browser and choose **Delete** . A dialog box displays confirming that you want to delete this item.

Deleting an item cannot be undone. You can also select the item and press the Del key for the same behavior. If you press Shift+Del the item is deleted without the confirmation dialog. The next item in the list is selected after an item is deleted. This means you can use the Shift+Del many times in a row to quickly delete many items for a specific type.

2.2.1.7. Accessing Submenus

To access a menu of relevant commands for a node, right-click the node in the Project Browser. An extended menu is often available by pressing **Shift**-right-click.

2.2.1.8. Scrolling in Windows

You can use your mouse scroll wheel/button in the AWR Design Environment platform windows in three scrolling modes:

- Standard: scrolling pans vertically
- **Shift**+scroll: scrolling pans horizontally
- **Ctrl**+scroll: scrolling zooms the display in and out

2.2.2. Creating, Opening, and Saving a Project

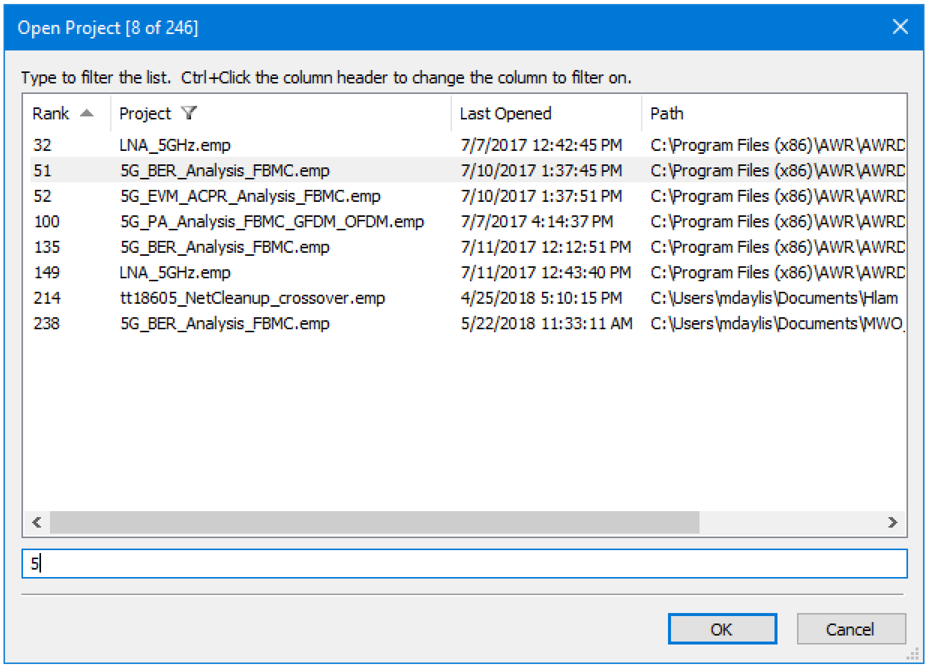
Creating a project is the first step toward building and simulating your designs.

When you start the program, a default empty project ("Untitled Project") opens. Only one project can be open at a time, although you can run more than one instance of the program. The name of the open project displays in the title bar.

To create a new project, choose **File > New Project**. Name the new project by choosing **File > Save Project As**. The project name displays in the title bar.

To create a new project with a foundry library, choose **File > New with Library** , then choose **Browse** to locate the **.ini* file for a specific foundry. The name of the foundry displays in the title bar. For more information about using foundry libraries see “Working With Foundry Libraries”.

To open an existing project, choose **File > Open Project** or **File > More Projects** to display the Open Project dialog box. When you start typing, the list is immediately filtered to display only those projects that match the text you type. You can filter the list by project name, use frequency (rank), date of last file opening, or file path.



To clear the current project from view, choose **File > Close Project**. You are prompted to save your changes, and the project is saved (if specified) and closed.

To save the current project, choose **File > Save Project**. The file is automatically compressed using a compression algorithm and saved as an **.emp* file.

2.2.2.1. Opening Example Projects

AWR software provides a number of project examples (**.emp* files) in the *C:\Program Files\AWR\AWRDE\17\Examples* or *C:\Program Files (x86)\AWR\AWRDE\17\Examples* directory to demonstrate key concepts, program functions and features, and show use of specific elements. You can filter project examples by keyword or search for an example by file name. A funnel icon in the column header indicates the column on which your search is filtered.

To search for and open a specific example project:

1. Choose **File > Open Example**.

   The Open Example Project dialog box displays with columns for the project name and keywords associated with each example project.

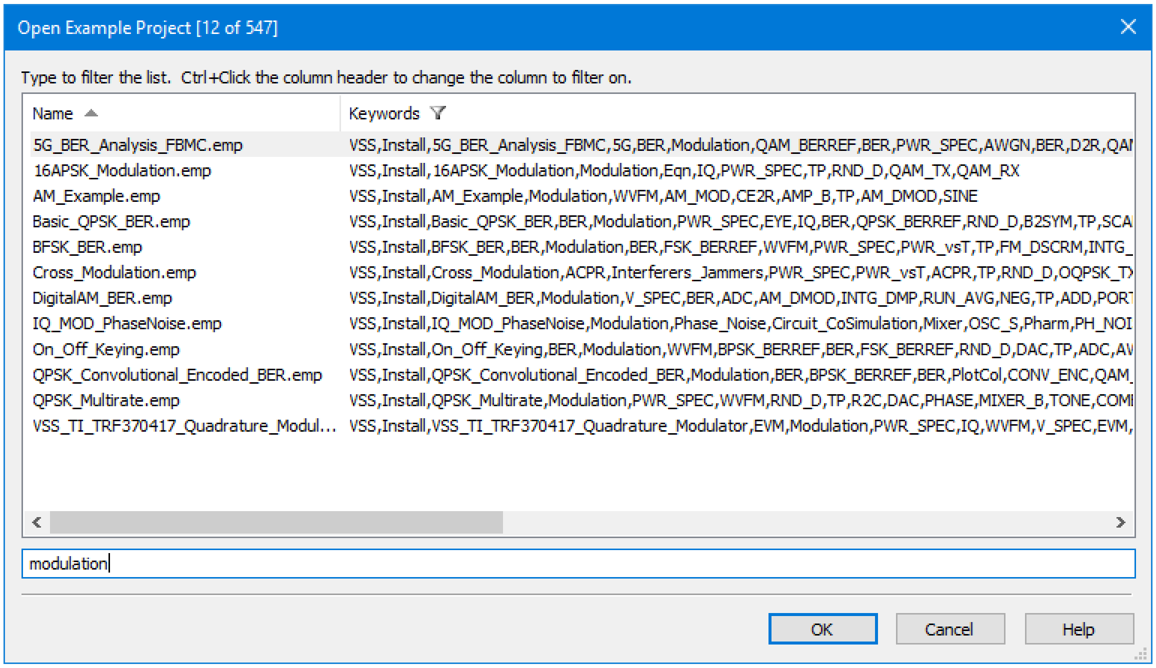
2. To filter the list by name, **Ctrl**-click the **Name** column header and begin typing an example project name in the text box at the bottom of the dialog box.

   The example list is filtered to display only those projects that match your input, or

3. To filter the list using a keyword, **Ctrl**-click the **Keywords** column header and begin typing a keyword in the text box at the bottom of the dialog box.

   The example list is filtered to display only those projects that have the input keyword associated with them.

For example, to list all example projects that include the keyword "modulation", **Ctrl**-click the **Keywords** column header if necessary and type "**modulation**". The list of project examples is filtered to display only those having the "modulation" keyword associated with them.



Filtering Examples

The Open Example Project dialog box filtering capability is quite powerful. The following are some tips for entering filters:

- Type part of a keyword and watch the matches filter as you type.
- Type part or all of a keyword, use a space and then type another word to filter both words. For example, if you type "mwo mixer" all the mixer examples for the Microwave Office program are listed.
- Use the "video" keyword to see all available videos.
- Use the "new" keyword to see all examples that are new or have new functionality added. Typing "new mwo" lists all new Microwave Office examples, and "new vss" lists all new VSS examples.
- Use the "mwo", "vss", or "ao" keywords to filter by products.
- Use the "install" keyword to see all examples in the program installation. Use the "web" keyword to see all examples in the Cadence AWR Knowledge Base.
- Use the "design_guide" keyword to see all examples set up as design guides or measurement templates.
- Use the "model_tester" keyword to see all examples set up to help you characterize specific types of models.
- Each example has additional keywords added. These keywords include simulator types (such as Cadence AXIEM® 3D planar EM or Cadence APLAC® HB), design types (such as amplifier or mixer), the unique measurements used in the example, and the unique models used in the example. For example, to locate examples that use a BIASTEE model, type "BIASTEE" to list all the examples.

For more information on this dialog box see "Open Example Project Dialog Box ".

2.2.2.2. Autosaving Projects

To automatically save your project and create backup files at set intervals:

1. Choose **Options > Environment Options**.
2. In the Environment Options dialog box, click the **Project** tab and select the **Autosave** check box.
3. Specify in **Minutes** how frequently you want to save your project.

The Autosave feature creates a backup file with an *.autobackup.emp* extension in the project file directory. Autosave automatically restores a project from the backup file if it detects that a project file closed without specifying Yes or No to the prompt to save it.

You can also select the **Save before Simulating** check box to automatically save your projects before simulations.

2.2.2.3. Saving Project Versions

To automatically save multiple versions of your project:

1. Choose **Options > Environment Options**.
2. In the Environment Options dialog box, click the **Project** tab and select the **Save revisions** check box.
3. Specify in **Previous versions** the number of project versions you want to retain.

This feature allows you to save up to nine versions of your project on disk; one each time you save a project. Each successive version is saved with a file extension that represents its currency. For example, when *my_circuit.emp* is the current version, *my_circuit.emp.bk1* is the previous version, *my_circuit.emp.bk2* is the version saved before it, and so on.

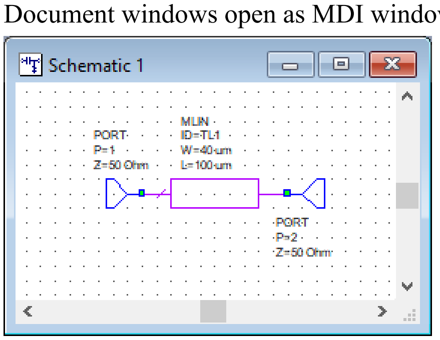
2.2.3. Displaying Document Windows

When you create a design in the AWR Design Environment platform, you create different types of documents such as schematics, layouts, and graphs. Each of these document types displays in its own window. You can double-click the item in the Project Browser to open its window. There are two types of windows:

- Multiple Document Interface (MDI) window: This window displays completely within the AWR Design Environment platform main window and is the default window type.
- Floating window: This window displays anywhere on the current computer display, including multiple monitors.

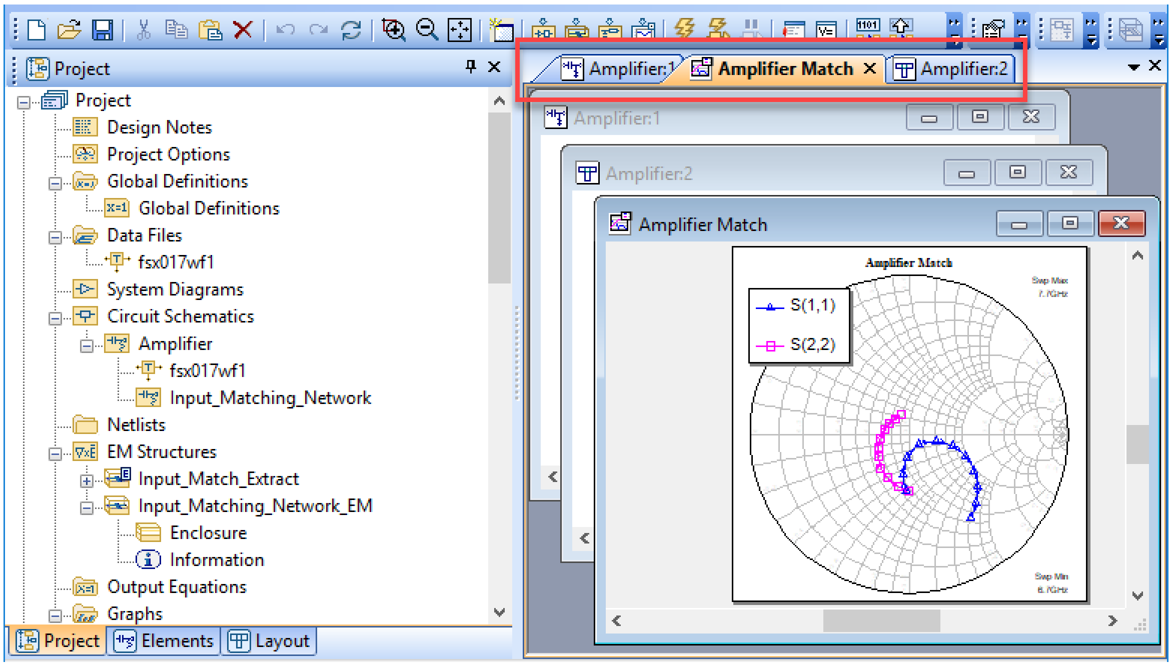
2.2.3.1. Multiple Document Interface (MDI) Windows

Document windows open as MDI windows by default, as shown in the following figure.



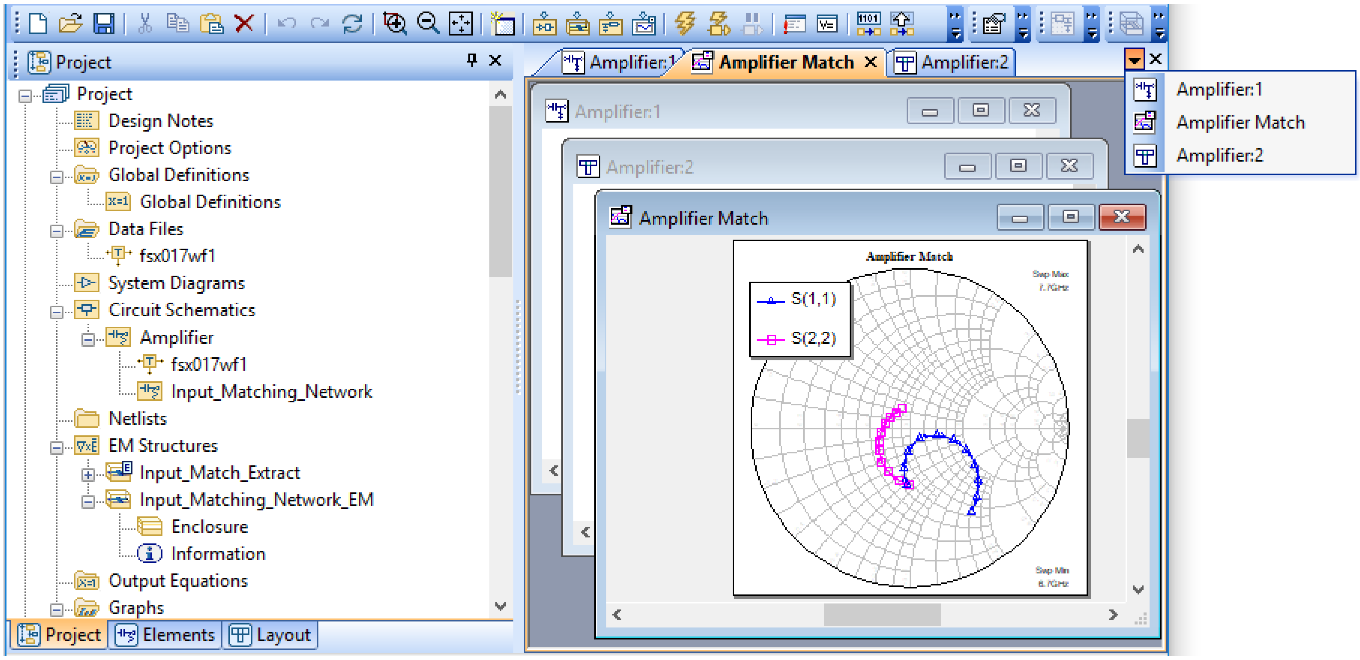
MDI windows have the following features:

- Controls on the upper right of the window title bar to minimize, maximize, or close the window.
- An icon on the upper left of the window title bar that indicates the document type. This icon can be double-clicked to close the window.
- A double-click of the title bar maximizes the window.
- A tab for each open MDI window displays at the top of the main AWR Design Environment platform window, as shown in the following figure. Tabs show at a glance all open windows and allow you to bring to the front any window that may be hidden behind other open windows.



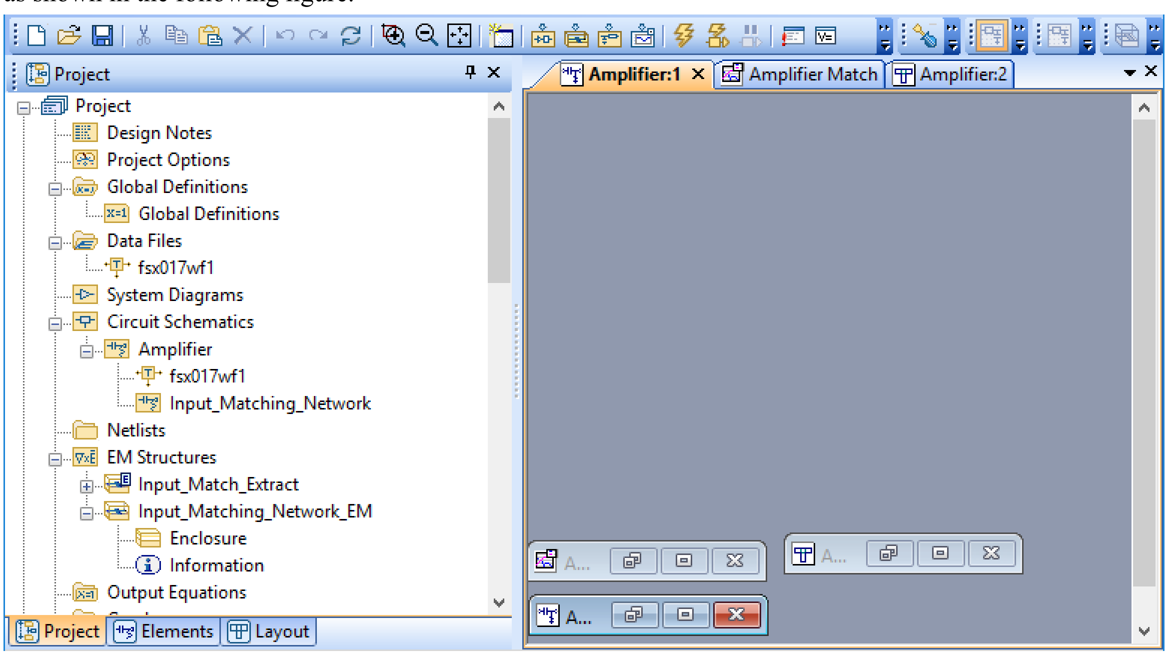
When you click on a tab to display the associated window, an "X" displays on the tab to allow you to close that window.

- At the far right side of the tabbed toolbar there are two additional controls. Click the "down arrow" for a list of all open windows, as shown in the following figure.

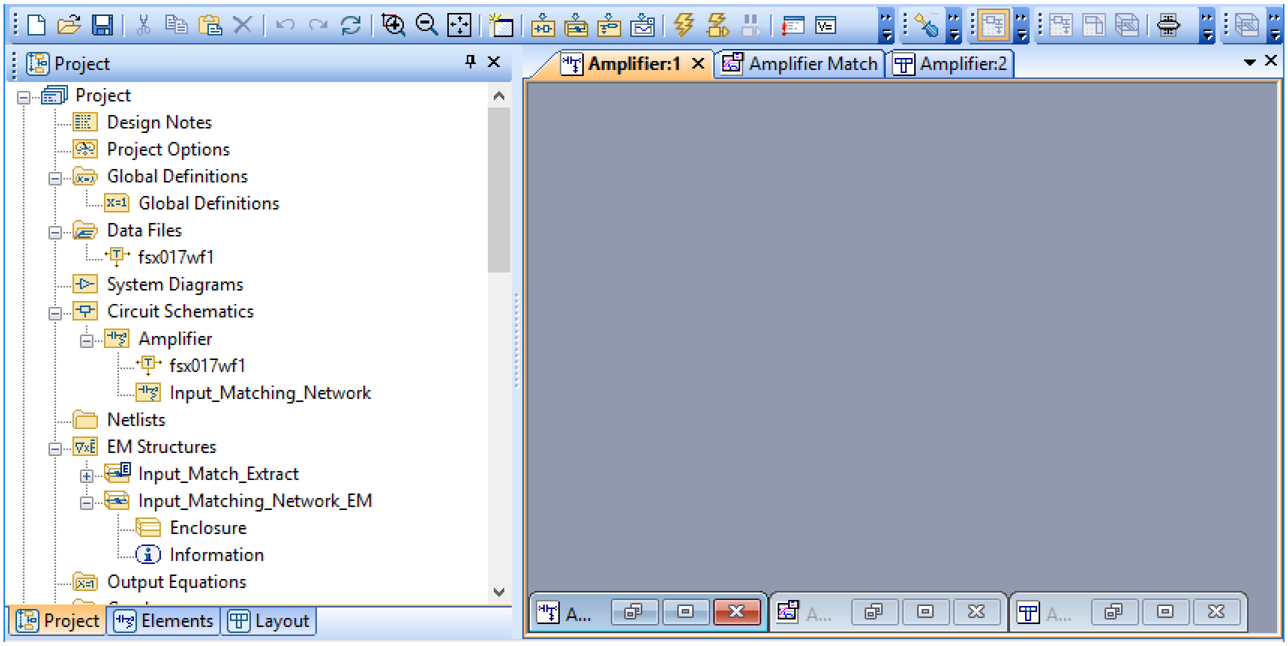


Click the X at the far right to close the currently active window.

- Pressing the **Ctrl + Tab** keys cycles through all open windows. Pressing **Shift + Ctrl + Tab** keys cycles in reverse order.
- The **Window** menu **Cascade**, **Tile Vertical** and **Tile Horizontal** display commands apply to MDI windows only.
- Minimized MDI window title bars display near the bottom of the AWR Design Environment platform main window, as shown in the following figure.



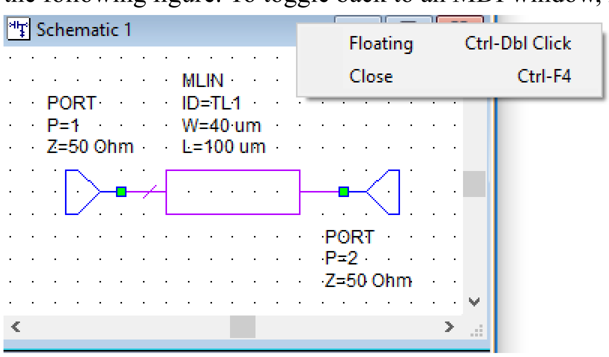
- Choosing **Window > Arrange Icons** reorganizes any minimized MDI windows. The following figure shows the minimized window title bars from the previous image rearranged.



- All of the commands in the Windows dialog box apply to MDI windows (to access this dialog box choose **Window > Windows**).

2.2.3.2. Floating Windows

You can change an MDI window to a floating window by right-clicking its title bar and choosing **Floating**, as shown in the following figure. To toggle back to an MDI window, repeat this action.



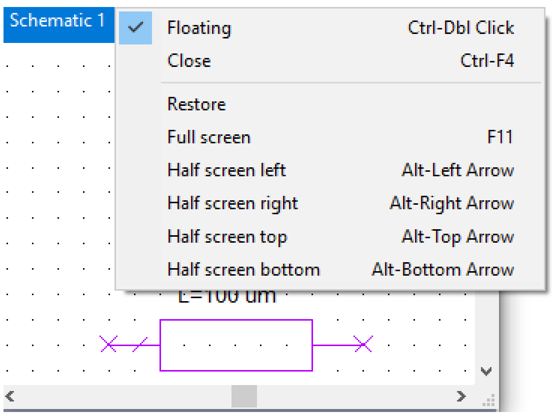
Alternatively, you can press the **Ctrl** key and double-click the window title bar to toggle between the MDI window state and floating window state. When switching between window states, the size and location of the window when it was last in that state is restored.

NOTE: Artwork cell windows are not restored to their previous size or MDI/floating state when you reopen them.

Floating windows have the following features:

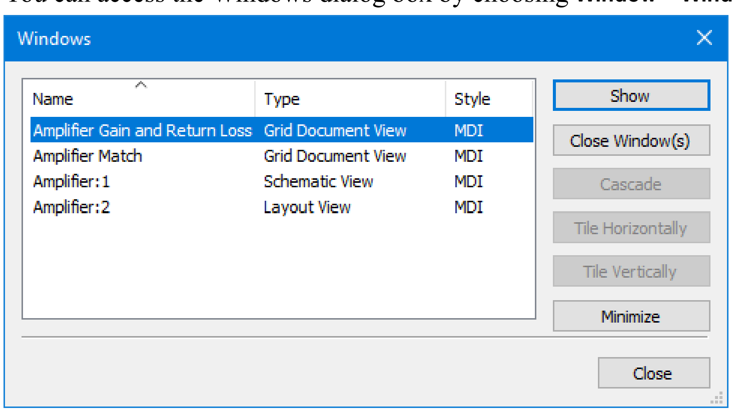
- Double-click the window's title bar to toggle between full screen and the previous size. If you press the **ALT** key while double-clicking the title bar in the center, the window is maximized. Double-click to the left of center to place the window on the left half of the screen, or double-click to the right of center to place the window on the right half of the screen.
- They always display on top of the main screen.

- Size and location are remembered when shutting down and reopening the program.
- Close the window by clicking the X icon in the upper right corner.
- When using the cascade or tiling commands (including commands in the Windows dialog box accessible by choosing **Window > Windows**), floating windows retain their current size and location.
- They hide when the main AWR Design Environment platform window is minimized.
- When changing the number of monitors in use (such as switching from two monitors to one monitor), the next time the floating window opens it is in the visible current display. You may need to first close the window from the Windows dialog box and then reopen the window.
- Right-click the window title bar to view options and hotkeys for resizing the window to full screen, or to the left, right, top, or bottom of the screen.



2.2.3.3. Windows Dialog Box

You can access the Windows dialog box by choosing **Window > Windows**, as shown in the following figure.



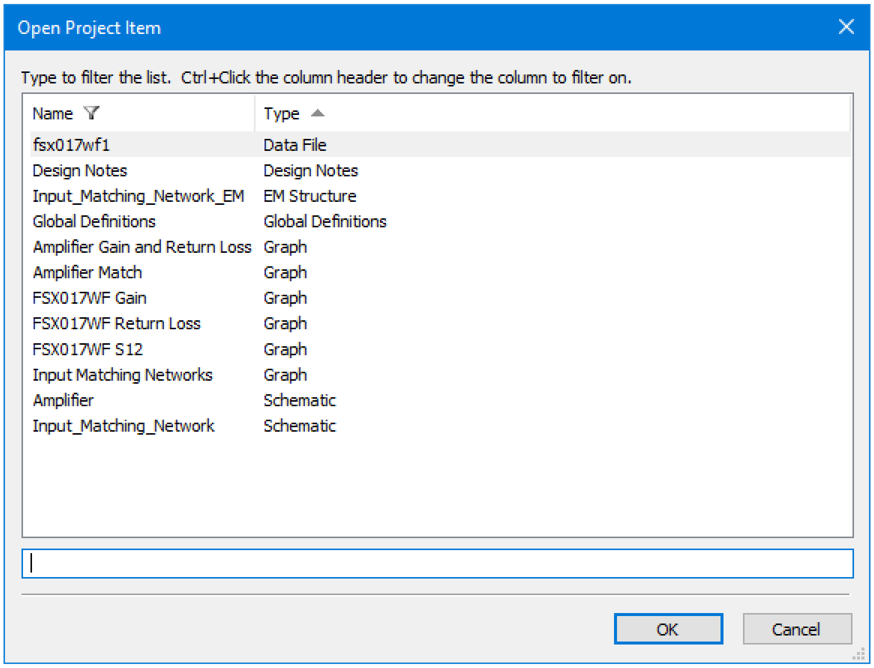
The Windows dialog box includes the following features:

- Clicking on either column header sorts by that column.
- Pressing the **Ctrl** key and clicking on multiple items selects those items.
- Clicking the command buttons on the right performs that operation for the selected windows.
- Floating windows only respond to the **Show** and **Close Window(s)** buttons.

2.2.3.4. Open Project Item

In large projects, it can be difficult to locate specific project items in the Project Browser.

To open a specific project item (for example, a schematic, system diagram, data file, or output equation), **Shift**-right-click anywhere in the Project Browser and choose **Open Project Item**. An Open Project Item dialog box displays with a list of items included in the project.



Select the item you want to open and click **OK**.

The Open Project Item dialog box includes the following features:

- Clicking on either column header sorts by that column.
- Typing in the text field at the bottom of the dialog box filters the display based on your text.
- **Ctrl**-clicking a column header changes which column's text is used to filter.
- **Ctrl**-clicking multiple items selects those items. **Shift**-clicking selects a range of items.

2.2.4. .vin files

When you close a project, a *.vin* file with the same stem name as the project file is created. The *.vin* file contains information about which windows are open, including their size and location. It also contains information about the collapsed and expanded state of the Project Browser. When a project that has an accompanying *.vin* file is opened, the project interface set up when the project was closed is restored. If a *.vin* file does not exist, when a project is loaded no windows are opened and the entire design hierarchy in the Project Browser is expanded.

2.2.5. Saving Projects As Project Templates

The AWR Design Environment platform allows you to save any project as a project template.

A project template is essentially a project that is saved with its options, LPFs, artwork cells, design notes, global definitions, frequency, graph, and measurement information, but not its simulated documents (for example, EM structures, data files, netlists, system diagrams or schematics, or single source measurements). Project templates provide an easy method for specifying sets of graphs, measurements, and outputs that are independent of any schematics, EM structures or data files. This information can be used in other projects or to perform comparisons between various data files. Project templates also include all the options and format information for a project. When a project template is opened, the graphs, measurements, options and outputs associated with that project template are read into the project.

To save a project as a project template, choose **File > Save As**, and select **Project Template (*.emt)** from the **Save as type** drop-down list. The project template is saved as an **.emt* file.

To specify a path to a project template choose **Options > Environment Options** and click the **File Locations** tab. In **Default Project Template**, browse to the location of the desired template. Every time you open a new project, the designated template is used.

For examples of using project templates, see "Using Project Templates with Template Measurements".

2.2.6. Specifying Global Project Settings

All options accessible under the **Options** menu apply per project, except for **Environment Options**, which apply to all projects under the current user. Options prefaced with "Default" can be overridden on each type document (for example, circuit schematic or graph). The remaining sections discuss details of several common settings to consider.

You can specify global settings for the units used within all schematics in a project, and for the simulation frequency used by all simulations performed within a project. In addition, you can specify global interpolation settings to employ during simulations.

2.2.6.1. Configuring Project Units

When running the AWR Design Environment software with the Layout feature license, units are configured per-LPF:

1. Choose **Options > Drawing Layers**. The Select LPF file dialog box displays.
2. Select the desired LPF, then click **OK** to display the LPF Options dialog box.
3. Under the **General** folder in the left pane, click **Units** and specify the desired units, then click **OK**.

When running the AWR Design Environment software without the Layout feature license, global units are configured for the project:

1. Choose **Options > Project Options** to display the Project Options dialog box.
2. Click the **Global Units** tab.
3. Select the desired units for each item, then click **OK**. Note that you can choose to set all items to use base units by clicking the **Use Base Units** button.

2.2.6.2. Configuring Global Project Frequency

To modify global project frequencies:

1. Choose **Options > Project Options**. The Project Options dialog box displays. Click the **Frequencies** tab to specify global frequency values. See "Project Options Dialog Box: Frequencies Tab " for more information about the dialog box.
2. To specify a frequency sweep, enter values for **Start**, **Stop**, and **Step**. To specify a frequency point, select the **Single Point** check box, and enter a **Point** value.
3. Click **Apply** and then **OK**.

You can always override global project frequency settings for a particular schematic, system diagram, netlist, or EM structure by specifying a local frequency. You do this in the Project Browser by right-clicking the individual schematic, netlist, or EM structure, choosing **Options**, and then deselecting the **Use Project Defaults** check box on the **Frequencies** tab.

2.2.6.3. Configuring Global Interpolation Settings

To modify the global interpolation settings:

1. Choose **Options > Project Options**. The Project Options dialog box displays. Click the **Interpolation/Passivity** tab to specify global interpolation settings. The Project Options dialog box displays. See "Project Options Dialog Box: Interpolation/Passivity Tab " for more information about the dialog box.
2. Modify the settings as desired, and click **OK**.

2.2.7. Working With Foundry Libraries

Often the AWR Design Environment software is used with Process Design Kits (PDKs) from various foundries. See the Cadence website for available foundries, or contact your local sales manager.

AWR Design Environment platform projects store the name of the process library with the project, so when the project is opened, the library is loaded with the project and the library name displays in the program title bar. If the current PDK *.ini* file is missing, you are prompted to browse for a replacement.

2.2.7.1. Adding and Removing Process Libraries

You can create a new project with a process library by choosing **File > New With Library**. A list of previously used libraries displays, as well as a **Browse** option to allow you to locate a foundry **.ini* file on your computer, and an **AWR Example Libraries** option that provides a selection of sample libraries for Silicon, GaAs and PCB technologies.

Choose **File > New with Library > Purge** to remove libraries with invalid file paths from the list of available libraries, or you can also manually add or remove process library references by choosing **Project > Process Library > Add/Remove Library**. An Add/Remove Process Library dialog box displays with the name and path to the **.ini* file for all the foundry libraries stored for your project. You can have more than one process library loaded at once. You would use this method if:

- You started a project without a process library and need to use the process library models, layouts, or other.
- You are migrating from one version of a library to another.
- You did design work with multiple process design kits.

When manually adding a PDK to a project, the LPFs, Global Definitions documents, and Artwork Cell Libraries from the new PDK are imported into the project. If layout options of the new PDK do not match existing layout options in the project, a Layout Options Mismatch Warning dialog box displays with a list of the mismatched options. You should understand the implications of changing layout options for the added PDK. See "Layout Options Dialog Box: Layout Tab " for details.

The PDK *.ini* file also allows you to reference other PDK *.ini* files. This is useful if you want to reference one file but use all the information from various PDKs, as common in multi-technology designs. The general structure of a PDK *.ini* file for this format is:

```
[Foundry]
Name=Sample Project PDK
```

```
Description=My multipdk
Version=1.0.0.0

[Child Libraries]
C:\Program Files (x86)\AWR\Foundry\Foundry1\Process1\1.0.1.0\process1.ini
C:\Program Files (x86)\AWR\Foundry\Foundry1\Process1\1.0.2.0\process2.ini
C:\Program Files (x86)\AWR\Foundry\Foundry2\Process1\1.1.0.0\process1.ini
```

2.2.7.2. Using Multiple PDKs and Multiple Versions of a PDK

When you add a new schematic to a project that has more than one process library loaded, the New Schematic dialog box allows you to choose which library to associate with the schematic. The library name and version (if there are multiple versions loaded) display in the title bar of the schematic. The associated PDK name and version display under **Process Library** on the "Options Dialog Box: Schematic Tab". You cannot modify the PDK name once a library is associated with the schematic. You can change the library **Version** using the pull-down list, and if the LPF and Global Definitions settings are synchronized with the initial library version setting, they auto-update when the PDK version changes. The PDK displayed in the schematic options is the primary library used by the schematic. Electrical models from other process libraries can be used in the schematic, but if two libraries loaded in the project contain a model with the same name, the schematic uses the one defined in its primary library. The same applies to layout cells used in the layout associated with the schematic, and to schematic symbols—preference is given to cells and symbols defined in the primary process library.

The AWR Design Environment platform does not prevent the schematic from using elements from other libraries. You can freely copy and paste elements between schematics, regardless of library association. The one restriction is that you cannot drag from the Elements Browser an element from a particular version of a library onto a schematic that is associated with another version of the library. Similarly, the **Replace Selected Element** command will not replace an element in a schematic associated with one version of a library with an element from another version of the same library.

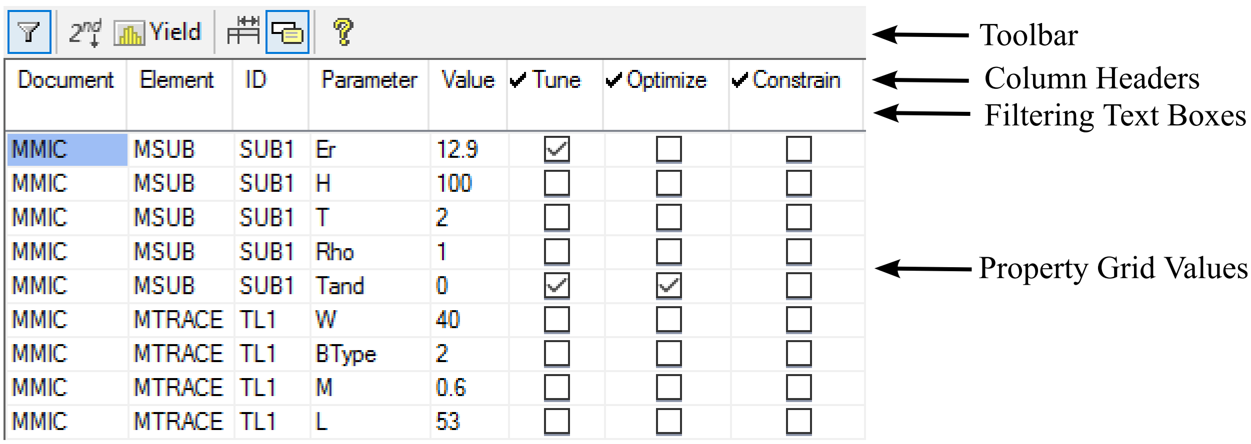
Normally when a project that uses a process library is opened, the AWR Design Environment platform checks to see if there is a newer version of the library available and asks whether to switch to the newer one. However, if a project uses multiple versions of a library, then this check is not performed. The AWR Design Environment platform will continue to use whichever versions the project calls for. You can manually change the versions through the Add/Remove Process Library dialog box.

2.3. Using Property Grids

Property grids are commonly used for organizing and editing values in a design. The following sections document common use of the grids; your use may be customized. Property grids are used in the:

- Variable Viewer - "Variable Browser"
- Element Properties - "Element Options Dialog Box: Parameters Tab"
- Layout Manager - "Drawing Layer Pane"

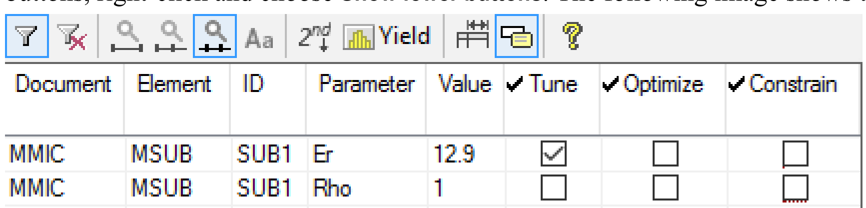
The property grid includes the following components:



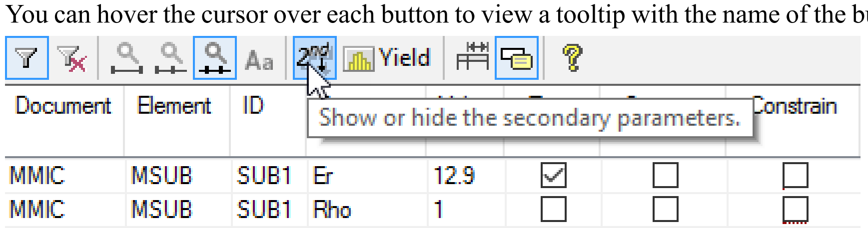
2.3.1. Property Grid Toolbar

Property Grid toolbar buttons control the display and content of the property grid. Most property grids use the buttons described in the following sections. Buttons particular to specific property grids are described in those sections.

To display additional toolbar buttons, right-click on the toolbar and choose **Show more buttons**. To hide the additional buttons, right-click and choose **Show fewer buttons**. The following image shows the common toolbar buttons.

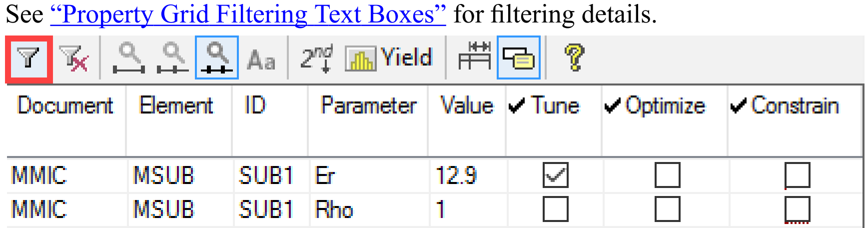


You can hover the cursor over each button to view a tooltip with the name of the button as shown in the following figure.

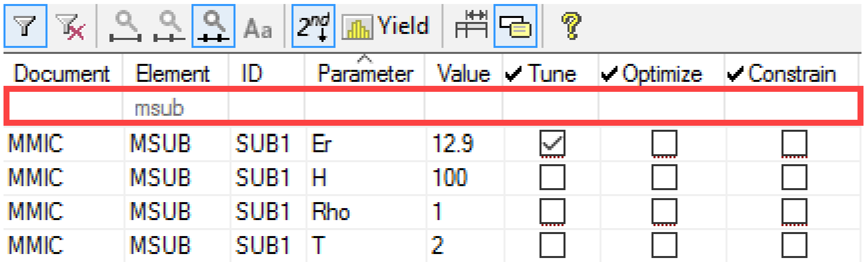


2.3.1.1. Button: Show the list filtered or unfiltered

This button toggles property grid filtering on or off. Click it to display a row of blank filtering text boxes under the column headers in the dialog box. Text that you type in the text box under a column filters the content of that column. See "Property Grid Filtering Text Boxes" for filtering details.

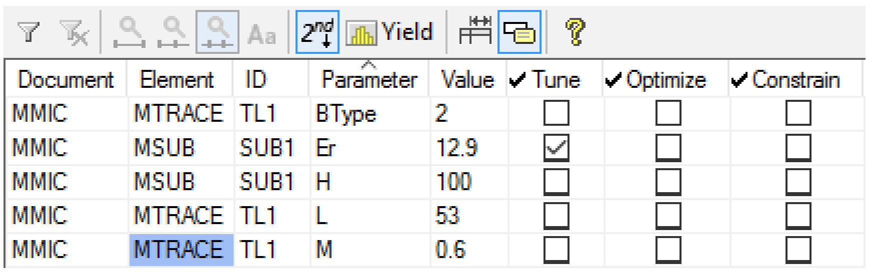


For example, typing "msub" in the filter row of the "Element" column provides the following result:



Document	Element	ID	Parameter	Value	✔Tune	✔Optimize	✔Constrain
	msub						
MMIC	MSUB	SUB1	Er	12.9	☑	☐	☐
MMIC	MSUB	SUB1	H	100	☐	☐	☐
MMIC	MSUB	SUB1	Rho	1	☐	☐	☐
MMIC	MSUB	SUB1	T	2	☐	☐	☐

When filtering is off:

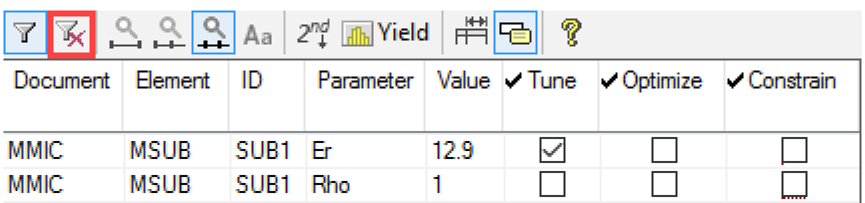


Document	Element	ID	Parameter	Value	✔Tune	✔Optimize	✔Constrain
MMIC	MTRACE	TL1	BType	2	☐	☐	☐
MMIC	MSUB	SUB1	Er	12.9	☑	☐	☐
MMIC	MSUB	SUB1	H	100	☐	☐	☐
MMIC	MTRACE	TL1	L	53	☐	☐	☐
MMIC	MTRACE	TL1	M	0.6	☐	☐	☐

- The row of blank filtering text boxes does not display.
- Previous filtering results no longer display and the column shows all items (in this example, elements other than "msub" display again).
- The additional filtering buttons to the right of this button are disabled.

2.3.1.2. Button: Clear the filters from all columns

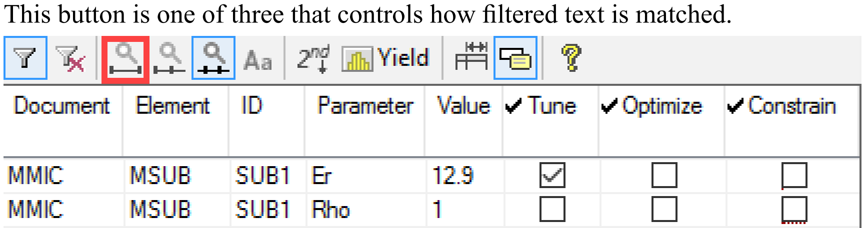
This button clears any text typed in one or more filter text boxes. It is only enabled when the "Show the list filtered or unfiltered" button is active.



Document	Element	ID	Parameter	Value	✔Tune	✔Optimize	✔Constrain
MMIC	MSUB	SUB1	Er	12.9	☑	☐	☐
MMIC	MSUB	SUB1	Rho	1	☐	☐	☐

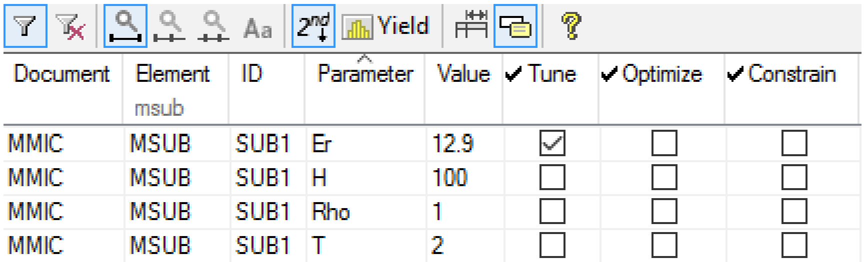
2.3.1.3. Button: Show values that match the text

This button is one of three that controls how filtered text is matched.

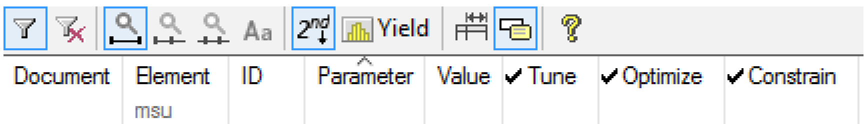


Document	Element	ID	Parameter	Value	✔Tune	✔Optimize	✔Constrain
MMIC	MSUB	SUB1	Er	12.9	☑	☐	☐
MMIC	MSUB	SUB1	Rho	1	☐	☐	☐

In this mode, the text you type must exactly match the text in the column below. For example, typing "msub" in the filter row of the "Element" column provides the following result:

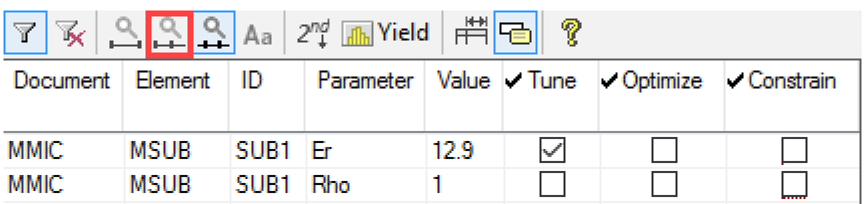


Typing "msu" provides the following result since there are no exact matches in the "Element" column.

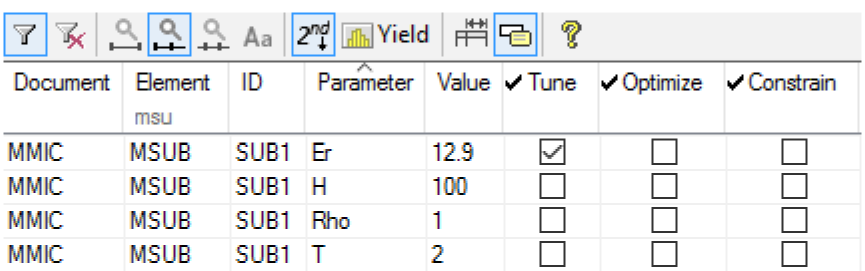


2.3.1.4. Button: Show values that start with matching text

This button is one of three that controls how filtered text is matched.

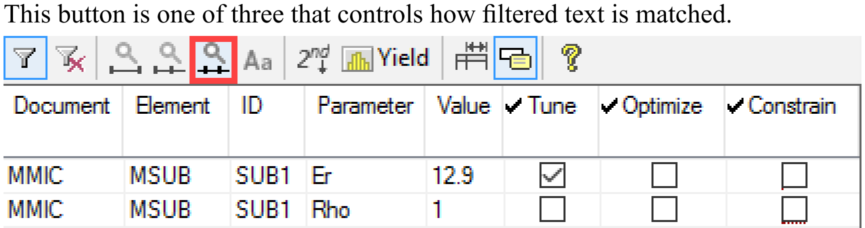


In this mode, the text you type must match the initial letter and subsequent letters in the column below; it does not have to match items exactly. For example, typing "msu" provides the following result since these letters match the first three letters of the "MSUB" items in the "Element" column.

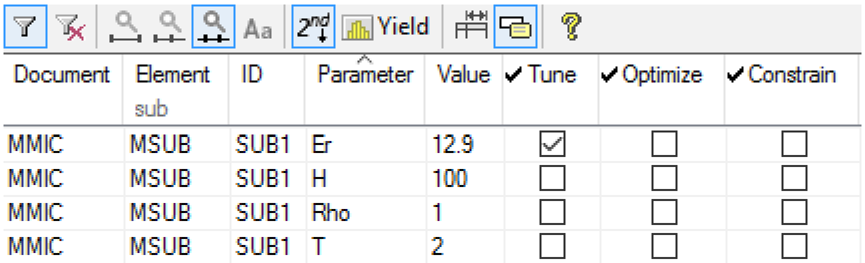


2.3.1.5. Button: Show values that contain matching text

This button is one of three that controls how filtered text is matched.

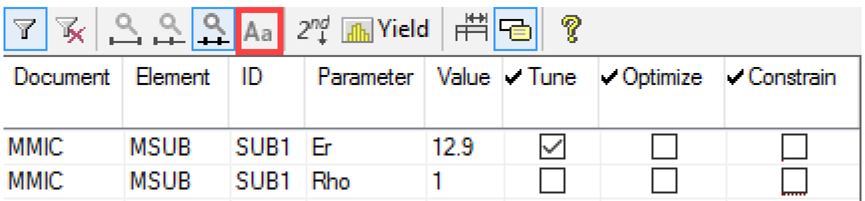


In this mode, the text you type can match any part of the text in the column below; it does not have to match items starting with the first letter. For example, typing "sub" provides the following result since these letters are included in the "MSUB" items in the "Element" column.



2.3.1.6. Button: Match case

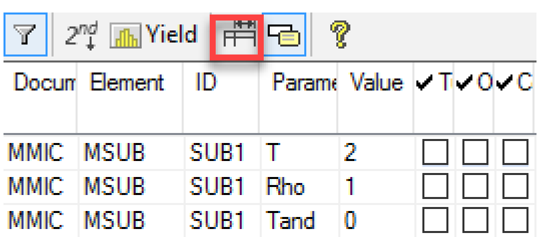
This button determines if the filter text is case sensitive.



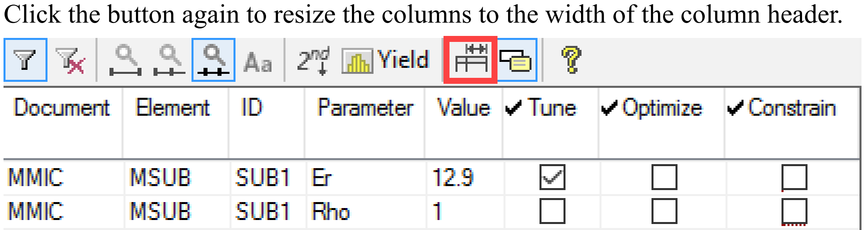
In all of the previous examples the text you typed is lowercase ("msub", "msu" and "sub") so none of these would provide a matched result with the uppercase "MSUB" in the "Element" column.

2.3.1.7. Button: Size the columns to the width of the text

This button adjusts each column width to the longest string found in each column, which helps fit more columns in the visible area.

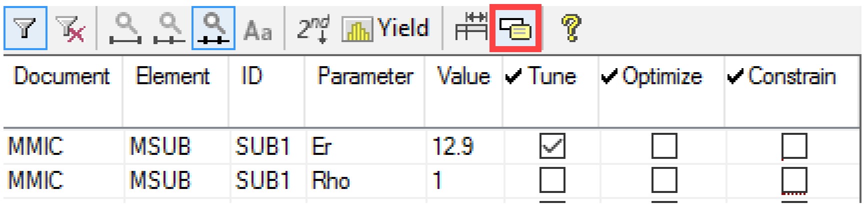


Click the button again to resize the columns to the width of the column header.

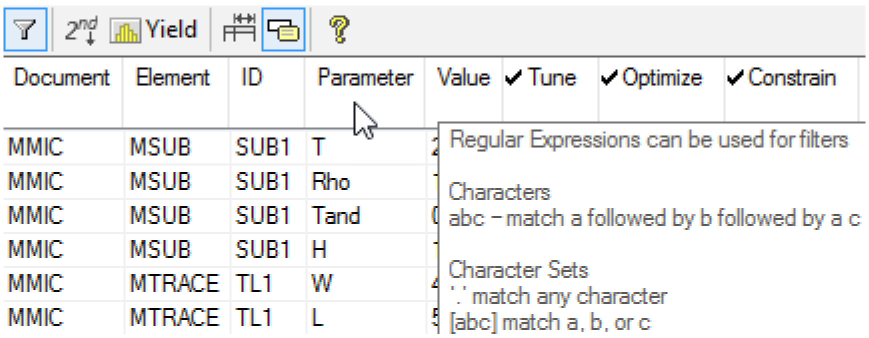


2.3.1.8. Button: Enable/Disable edit tool tips

This button toggles on or off the display of helpful tooltips for the filtering text boxes.

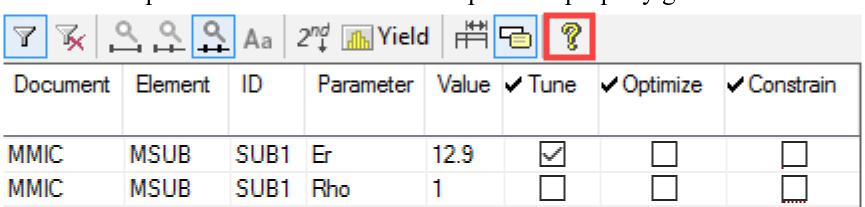


Click it to display popup filtering tooltips when you hover the cursor over the filter text boxes under each column.



2.3.1.9. Button: Show Help on using this window

This button opens the associated online Help for the property grid.

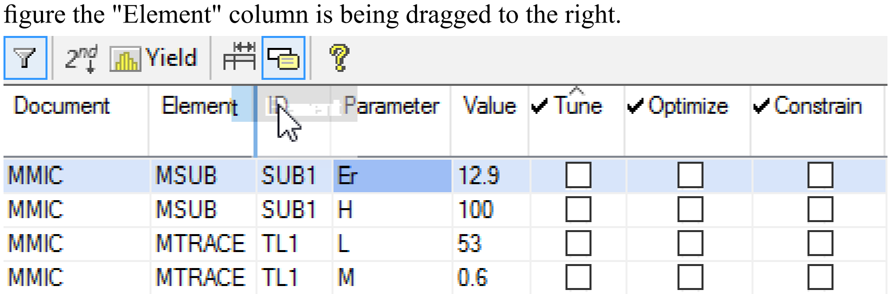


2.3.2. Property Grid Column Headers

Column headers describe the content of each column in the property grid. There are various ways to control how the columns and their contents display.

2.3.2.1. Changing Column Order

Click a column header and drag left or right across the property grid to move the column. For example, in the following figure the "Element" column is being dragged to the right.



After releasing the mouse button the column order is updated, with the "Element" column now to the right of the "ID" column.

Document	ID	Element	Parameter	Value	✔ Tune	✔ Optimize	✔ Constrain
MMIC	SUB1	MSUB	Er	12.9	☐	☐	☐
MMIC	SUB1	MSUB	H	100	☐	☐	☐
MMIC	TL1	MTRACE	L	53	☐	☐	☐
MMIC	TL1	MTRACE	M	0.6	☐	☐	☐

2.3.2.2. Changing Column Size

To change the size of a column, click and drag on the bar between columns to increase or decrease column size. Notice that the cursor display changes while dragging.

Document	ID	Element	Parameter	Value	✔ Tune	✔ Optimize	✔ Constrain
MMIC	SUB1	MSUB	Er	12.9	☐	☐	☐
MMIC	SUB1	MSUB	H	100	☐	☐	☐
MMIC	TL1	MTRACE	L	53	☐	☐	☐
MMIC	TL1	MTRACE	M	0.6	☐	☐	☐

2.3.2.3. Optimizing Column Size

Resize a column to the widest text in that column by holding the cursor over a column divider. When the cursor display changes, double-click to resize that column.

2.3.2.4. Sorting Rows of a Column

Click a column header to sort the property grid by that column. The first click sorts the column in ascending order. In the following figure, the chevron symbol at the top of the "Parameter" column indicates that items in this column are sorted in ascending order.

Document	ID	Element	Parameter	Value	✔ Tune	✔ Optimize	✔ Constrain
MMIC	TL1	MTRACE	BType	2	☐	☐	☐
MMIC	SUB1	MSUB	Er	12.9	☐	☐	☐
MMIC	SUB1	MSUB	H	100	☐	☐	☐
MMIC	TL1	MTRACE	L	53	☐	☐	☐

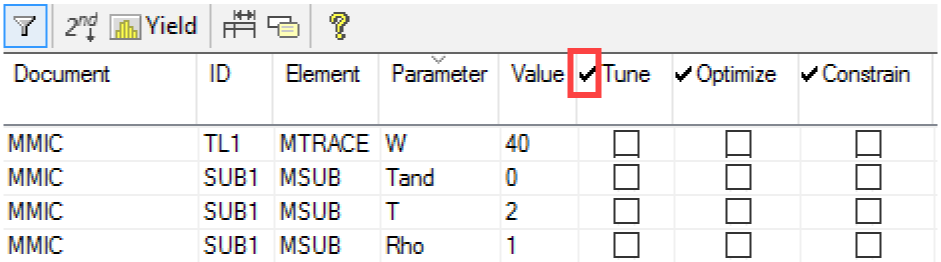
The second click sorts the column in descending order. In the following figure, the inverted chevron symbol at the top of the "Parameter" column indicates that items in this column are sorted in descending order.

Document	ID	Element	Parameter	Value	✔ Tune	✔ Optimize	✔ Constrain
MMIC	TL1	MTRACE	W	40	☐	☐	☐
MMIC	SUB1	MSUB	Tand	0	☐	☐	☐
MMIC	SUB1	MSUB	T	2	☐	☐	☐
MMIC	SUB1	MSUB	Rho	1	☐	☐	☐

Some columns support a third click which returns the column to the original, unsorted order.

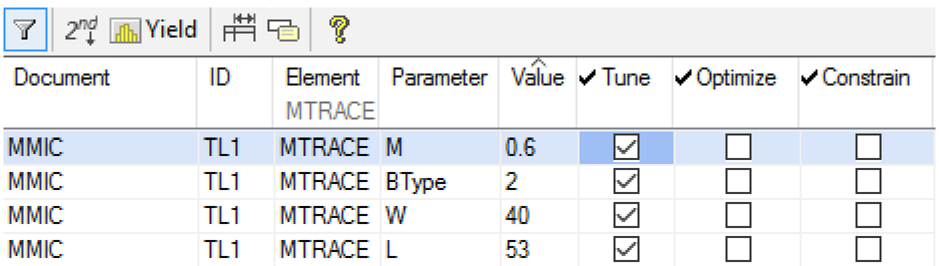
2.3.2.5. Selecting All/Nothing in a Column

A column that contains check boxes may include a small check mark icon to the left of the column header. Click on this icon to toggle between selecting every item in that column or selecting no items in the column.



2.3.3. Property Grid Filtering Text Boxes

The row of blank filtering text boxes allows you to filter the property grid by columns. To filter on a column, click in the (empty) filter text box below that column header and type the text you want to filter for in that column. For example, to find all microstrip lines in your project you can type "MTRACE" in the filter text box below the "Element" column as shown in the following figure.

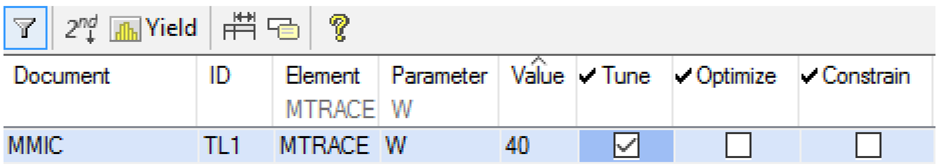


Filter text boxes also support regular expressions, which allow you to perform intelligent searches. The form and functionality of these regular expressions is modeled after the regular expression facility in the Perl 5 programming language. The following table shows some syntax examples:

Syntax	Comment
.	Match any single character.
*	Match zero or more of the preceding characters.
+	Match one or more of the preceding characters.
?	Match zero or one of the preceding characters.
!	Filter out subsequent characters.
\d	Match any digit (0-9).
[ch]at	Match cat and hat.
W[1-3]	Match W1, W2, and W3.
MBEND|MLIN	Match MBEND or MLIN.
^M	Match names that start with M.
^W\d+	Match names that start with W followed by one or more digits.

Syntax	Comment
\$$	Match names that end in $.
ID=TL\d	Match names that contain ID=TL followed by a digit.

You can apply a second filter to the results of the first search or filter on multiple columns by adding filter text for each column. Using the previous example, to see the width parameter for every microstrip line in a project you can type "MTRACE" in the filter text box in the "Element" column and then type "W" in the filter text box for the "Parameter" column. You can extend this filtering to any number of columns.



You can also filter on check boxes by typing "1", "T", or "C" for checked and conversely "0", "F", and "U" for unchecked check boxes.

2.3.4. Property Grid Values

Property grid values are the rest of the data/items in the grid. The following sections describe various means of working with data in the property grid.

2.3.4.1. Changing Values

Select any text or numerical item and begin typing to enter new values.

2.3.4.2. Selecting/Clearing Check Boxes

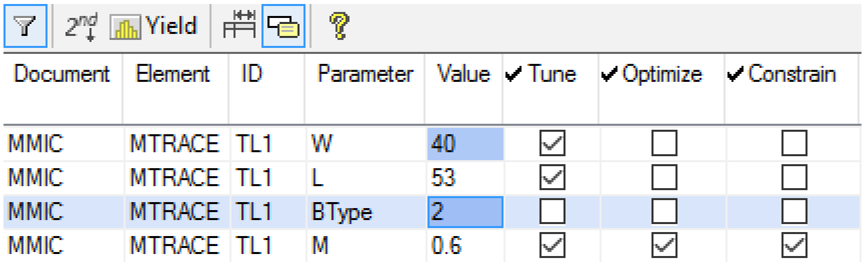
Select any check box to toggle the setting. Right-click on a check box to view the following options:

- Uncheck All But This (item): Only that row is selected, all others are cleared. Alternatively, **Alt**-clicking the check box does the same thing.
- Check All But This (item): All other rows except this row are selected.

You can also select or clear the check mark icon in the column header to select or clear the entire column.

2.3.4.3. Selecting Multiple

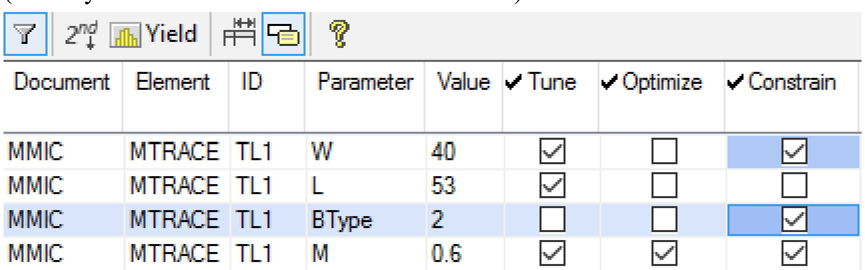
You can select multiple items in the same row by **Ctrl**-clicking each item to toggle its selection on or off. **Shift**-clicking items selects all the rows between the first row you select and the last row you select. Selected items display in a darker color while the entire row with the most recently selected item in it displays in a lighter color. In the following example the text "40" was first selected, then by **Ctrl**-clicking, the text "2" was selected in the "Value" column.



Document	Element	ID	Parameter	Value	✔ Tune	✔ Optimize	✔ Constrain
MMIC	MTRACE	TL1	W	40	☑	☐	☐
MMIC	MTRACE	TL1	L	53	☑	☐	☐
MMIC	MTRACE	TL1	BType	2	☐	☐	☐
MMIC	MTRACE	TL1	M	0.6	☑	☑	☑

When multiple items are selected, changing the value of one of the items changes it for all of the selected items.

If you click on another column in the row in which you made your last selection, the same column item is selected in the previously selected row. For example, in the previous figure, if you click in the "Constrain" column in the last active row (where you selected the "2" in the "Value" column), the item in the "Constrain" column of the prior active row (where you selected "40" in the "Value" column) is also selected.



Document	Element	ID	Parameter	Value	✔ Tune	✔ Optimize	✔ Constrain
MMIC	MTRACE	TL1	W	40	☑	☐	☑
MMIC	MTRACE	TL1	L	53	☑	☐	☐
MMIC	MTRACE	TL1	BType	2	☐	☐	☑
MMIC	MTRACE	TL1	M	0.6	☑	☑	☑

In addition, you can move a selection from one column to another using the left and right arrow keys, or press **Ctrl+A** to select all items in a column.

2.4. Organizing a Design

You can logically organize designs in the AWR Design Environment platform.

- Window-in-window: You can place a view of a window into another window.
- User Folders: You can build a folder structure and put any item in these folders.

2.4.1. Window-in-Window

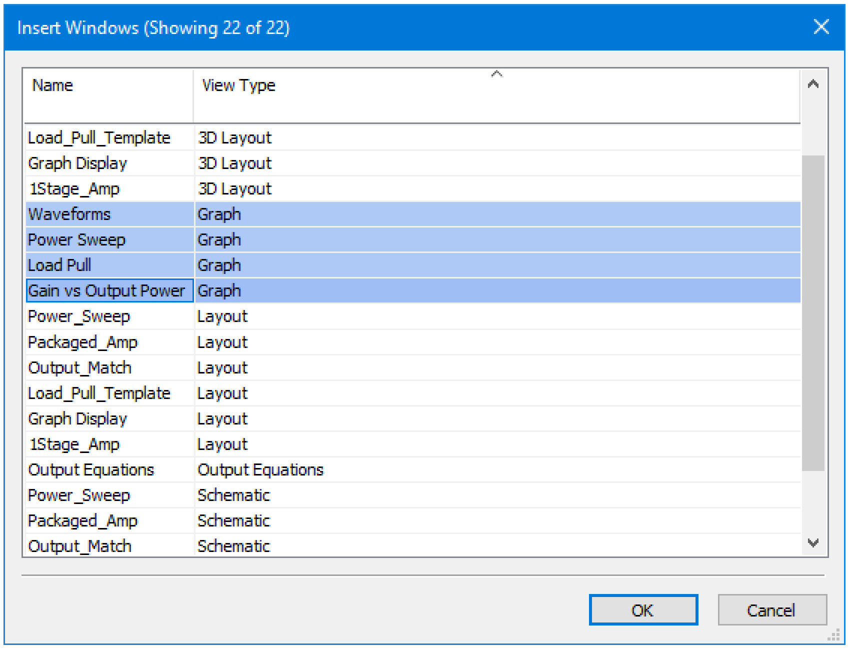
The AWR Design Environment software allows you to embed a view of a document into another document. For example, in a schematic window you can insert different graphs showing the schematic simulation results. Documents, including Global Definitions windows, can contain other live schematics, system diagrams, graphs, Output Equations, layouts and 3D layout views. This capability allows you to build design reports containing various views.

2.4.1.1. Inserting a Window-in-window

To add multiple Window-in-windows:

1. Make the host document (for example, the schematic or output equation document) the active window.
2. Choose **Draw > Insert Windows** or click the corresponding button on the toolbar.
3. The Insert Windows dialog box displays with a list of supported views you can insert as a Window-in-window in the host document. You can sort the views by Name or View Type by clicking on the column header. You can filter on the Name or View Type by typing in the first row under the header. Select a view for insertion as a Window-in-window

by clicking on it. The selected view is highlighted as shown in the following figure. Hold **Shift** or **Ctrl** to multi-select views for insertion. Click **OK** to close the dialog box.



4. If only one view is selected for insertion, it is instantly placed into the host document. If multiple views are selected the Align Shapes to Array dialog box opens with options for controlling windows placement, as described in [“Aligning Window-in-windows”].

5. After placement, you can rearrange Window-in-window objects with other objects that support the **Draw** menu **Align Shapes** and **Make Same Size** commands.

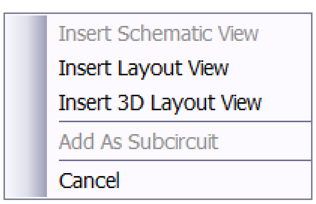
2.4.1.2. Adding Window-in-window from the Project Browser

You can also add a Window-in-window by dragging it from the Project Browser:

1. Make the target document (for example, the schematic or graph) the active window.

2. In the Project Browser, click and drag the item you want to add to the target document to the target document window in the workspace.

3. When adding a schematic window to another window, right-click to perform this operation and display a menu on the target window that allows you to add the schematic as a schematic, a subcircuit, or as the layout (or 3D layout) view of the schematic.



4. Release the mouse button to display a special cursor with a rectangle, then click and drag to define the rectangular area into which you want the added object to display. Grid snapping is active to assist with window sizing.

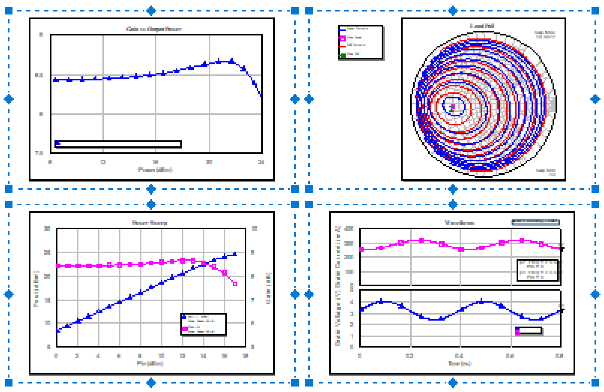
5. Release the mouse button to view your window within a window. You can select the window to move it by dragging, drag its selection handles to resize it, or press **Delete** to delete it.

2.4.1.3. Editing Window-in-Window

To activate the window contents for editing, double-click a window or right-click it and choose **Activate View**. An active window displays a border. To deactivate a window right-click the window and choose **Deactivate View** or just click outside the embedded view.

2.4.1.4. Aligning Window-in-windows

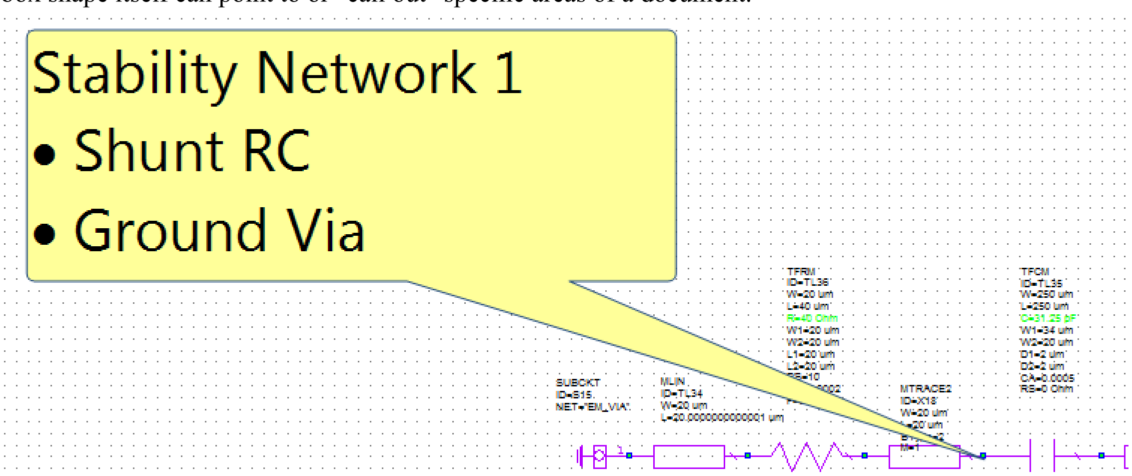
To array a group of windows **Shift**-click the windows to multi-select them, then choose **Draw > Align Shapes > Space and Size as Array** or the corresponding toolbar button. In the Align Shapes to Array dialog box, set the array order and spacing parameters. The first window in the selection list is the anchor element of the array, and all other windows are resized to the same height and width of the anchor window. See "Align Shapes to Array Dialog Box " for details. The following figure shows the results of the array operation.



You can align and resize windows by choosing commands in the **Draw > Align Shapes** and **Draw > Make Same Size** menus. Window-in-window objects can also be aligned or resized with other objects that support the align or resize commands. In general the first object you select serves as the anchor object, and other objects are aligned to it, or resized to match it.

2.4.2. Rich Text Boxes

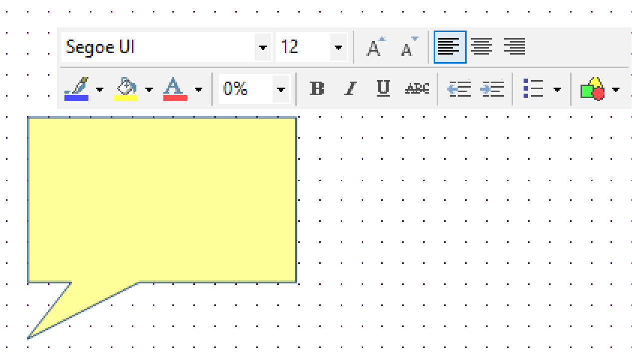
A rich text box is a graphical object for adding information to schematics, system diagrams, graphs, and equation pages. It does not function in Layout Views. It allows formatting of text fonts, colors, sizes, bullets, and boundaries. The text box shape itself can point to or "call out" specific areas of a document.



2.4.2.1. Adding Rich Text Boxes

To add a rich text box, choose **Draw > Rich Text** or click the **Text Box** button on the Graph toolbar.

In the document you want to annotate, click and drag to create a text box, releasing the mouse button when it reaches the desired size. The cursor flashes to indicate that you can type to enter the text.



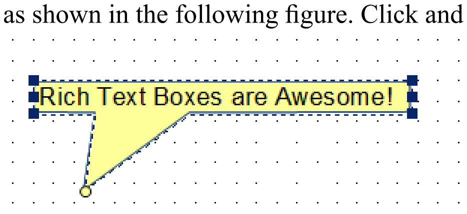
A control box displays for setting the text characteristics and position, and the box shape, color, and translucency. After typing, the height of the box auto-sizes to the text you add. Press **Enter** to add a new paragraph, and click outside the text box to exit editing mode.

2.4.2.2. Editing Rich Text Boxes

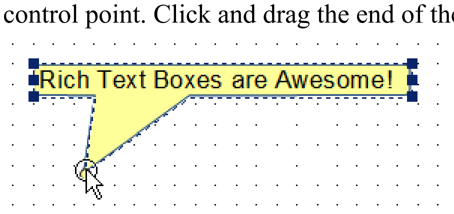
There are several editing modes for rich text boxes:

- Click on the box to view controls for editing the box size.
- Click and drag the mouse to move the text box.
- Double-click inside the text box to edit and format the text. The editing controls display above the box, and tool tips display when you hover the cursor over each control.

When you click on the text box to edit the size, edit controls display on the corners and mid-point of its vertical edges as shown in the following figure. Click and drag one of these controls to adjust the text box size.



Depending on your shape selection, your text box may or may not have a "callout tail" (the arrow outside of the text box). The edit point on the end of the tail displays with a yellow dot and the cursor displays as a circle when over this control point. Click and drag the end of the tail to adjust its size, position, and angle outside the box.



The following commands apply to selected text only and also to the cursor location in the text box:

- Font name
- Font size
- Text color
- Bold
- Italics
- Underscore
- Strikeout

The following commands apply to the text in the paragraph where the cursor is located:

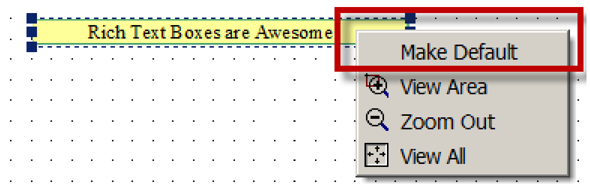
- Justification (left, right, center)
- Indent
- Outdent
- Bullets

The following commands apply to the drawing of the entire rich text box object:

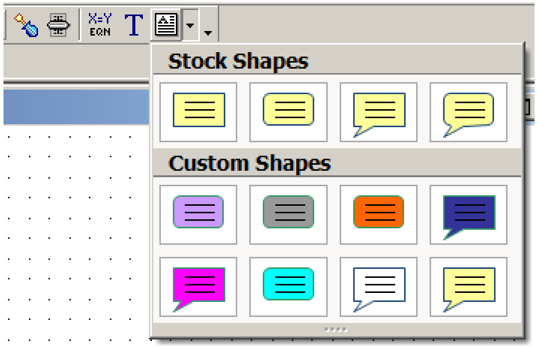
- Border color
- Fill color
- Translucency
- Shape

2.4.2.3. Saving Text Box Configurations

You can save the text characteristics and position and box style as a default for subsequent text boxes by right-clicking the text box and choosing **Make Default**.



The AWR Design Environment software saves your last eight configurations as defaults. When you add a text box from the toolbar, click the down arrow next to the button to view a drop-down menu of saved custom configurations.

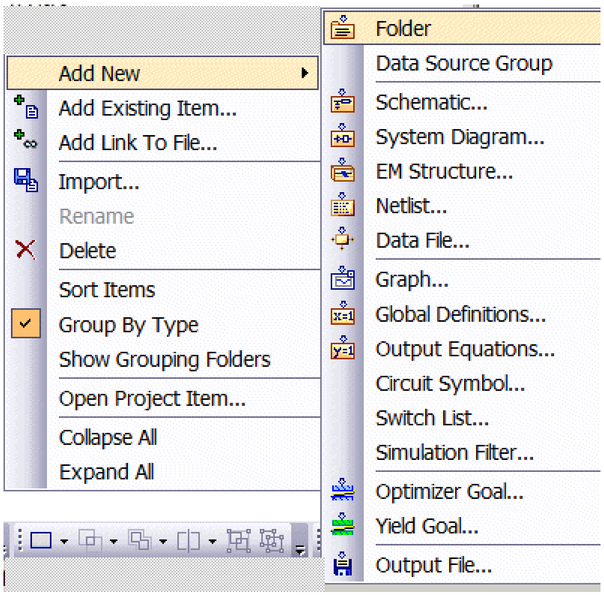


2.4.3. User Folders

The **User Folders** node in the Project Browser allows you to create a customized folder structure and then place any project item (for example, a schematic or graph) into those folders.

2.4.3.1. Adding User Folders

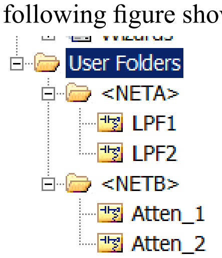
You can add a user folder by right-clicking the **User Folders** node in the Project Browser and choosing **Add New > Folder**. You can add subfolders beneath existing user folders with the same command.



When you add a new folder you can edit its default name.

Grouping Collections Networks as a Document Set

You can use User folders to define a Document Set by using a <name> convention. To create a network group folder, add a User folder and rename it using angle brackets, or right-click and choose **Add New > Data Source Group**. The following figure shows two collections of networks that represent different design alternatives for NETA and NETB.



A measurement on the data source group automatically generates a measurement on each document in the folder. See "Working with Data Source Groups" for details on using a data source group.

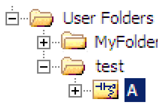
2.4.3.2. Renaming User Folders

You can rename a user folder by right-clicking the folder in the Project Browser and choosing **Rename**, or by pressing the **F2** hotkey.

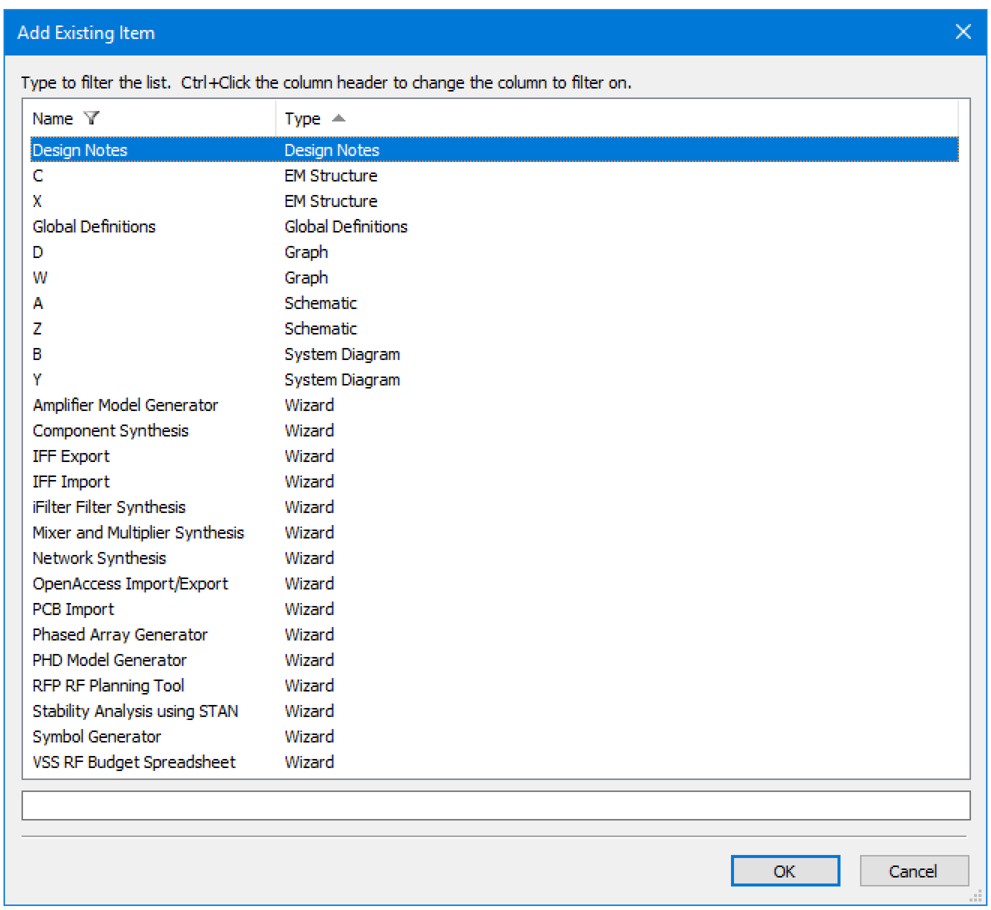
2.4.3.3. Adding Items to User Folders

You can add single items from the Project Browser by selecting an item and dragging it onto a user folder. The following figure shows a schematic being added (copied) to the "Test" folder.

After you release the mouse button the copied item displays in the folder.

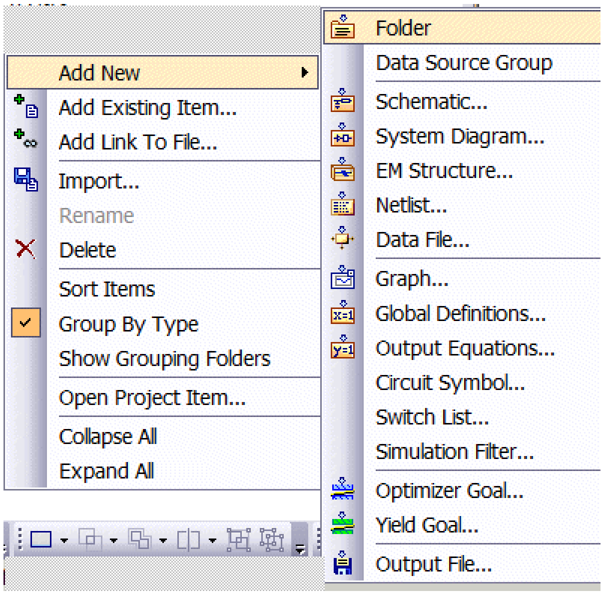


You can add multiple items to a user folder at one time. Select the destination folder, right-click and choose **Add Existing Item**.



In the Add Existing Item dialog box, **Ctrl**-click to add/remove multiple items. You can also **Shift**-click to add/remove a range of items. To filter the list of items, type into the text box at the bottom of the dialog box. All of the selected items are added to the selected destination folder.

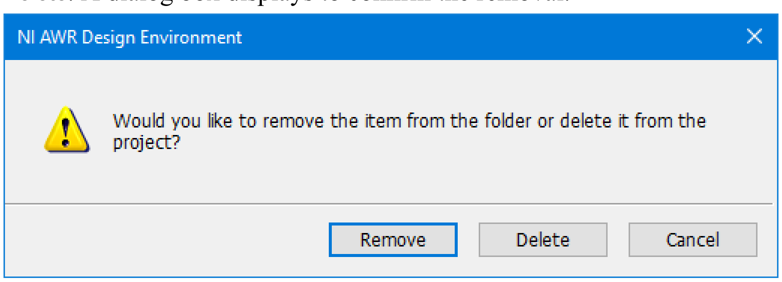
You can also add new items to a user folder. Right-click the desired folder and choose **Add New > <item>**, where <item> is the item you want to add.



The item is added to the selected folder as well as to the corresponding category folder in the Project Browser.

2.4.3.4. Removing Items from User Folders

You can remove items from folders by selecting the item and pressing the **Del** key, or by right-clicking and choosing **Delete**. A dialog box displays to confirm the removal.



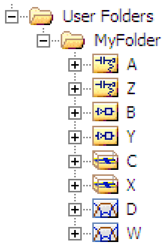
If you press **Shift + Del** the item is deleted from both the folder and project without prompting. If you press **Ctrl + Del** the item is deleted from ONLY the folder without prompting.

2.4.3.5. Moving Items in User Folders

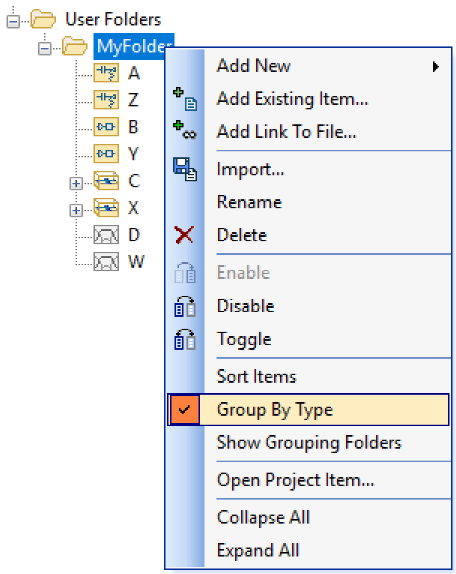
You can move an item to a different folder by selecting the item and dragging it from the source folder to the destination folder. To copy the item, press the **Ctrl** key during this operation.

2.4.3.6. Organizing Items in User Folders

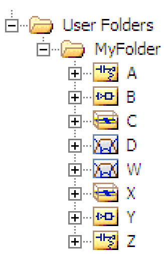
By default, the items in user folders are sorted by object type (for example by schematic or graph) and then alphabetically within the type. The following figure shows several items in a user folder.



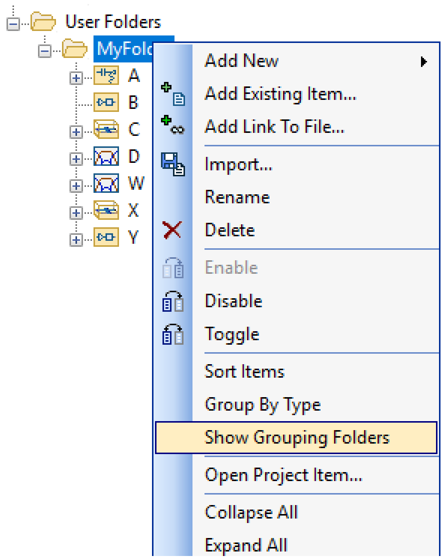
You can sort all items alphabetically (regardless of type) by right-clicking the folder and choosing **Group by Type** to toggle this option off.



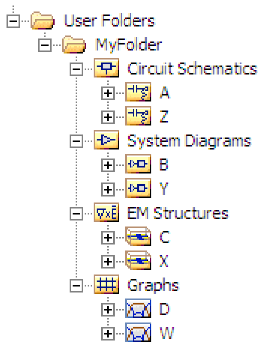
After this command, the example displays as follows.



You can organize the folders by object type by right-clicking the folder and choosing **Show Grouping Folders**.



After this command, the example displays as follows.



You can also manually change the order of items in a folder by selecting an item and pressing the **Alt** key and the Up or Down arrow keys to move the item up or down in the order. If you move items and want to reset the order, you can right-click the folder and choose **Sort Items**.

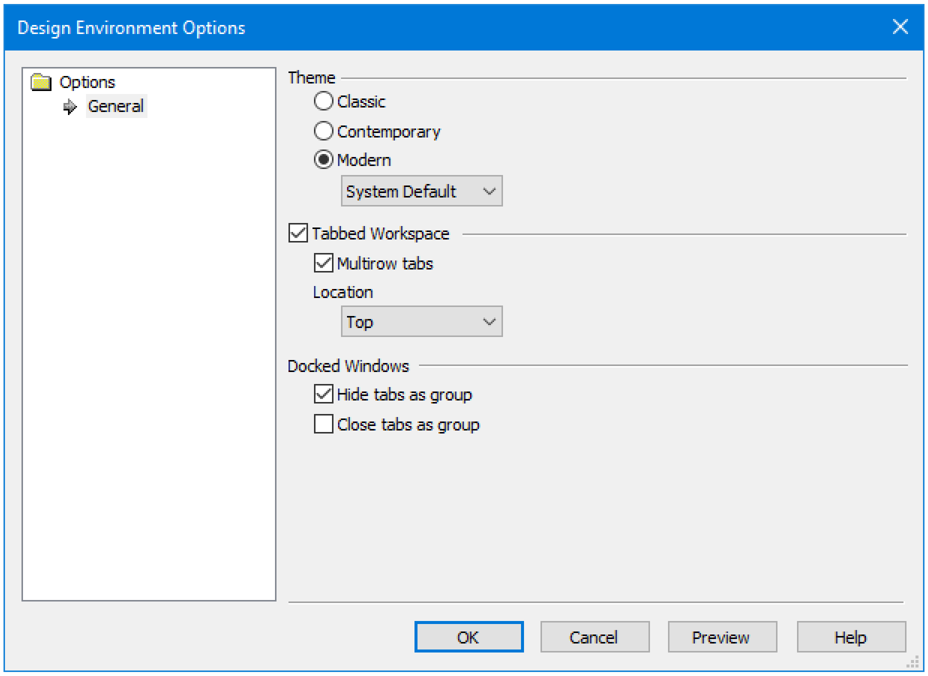
2.5. Customizing the Design Environment

The AWR Design Environment platform provides a number of ways to customize your design environment, including its appearance, tabbed workspace, dockable windows, hotkeys, menu and toolbar content, and use of scripts to automate repetitive tasks.

2.5.1. Customizing Workspace Appearance and Tabs

The Environment Options dialog box (**Options > Environment Options**) has a number of tabs with options that apply to every project you create. The most common settings are the **Save Options** on the **Project** tab for specifying project file saving options; all settings on the **Colors** tab for specifying schematic, layout, and other object colors; and all settings on the **File Locations** tab for specifying default directories.

The Design Environment Options dialog box (**Tools > Options**) contains options to customize the display of the design environment such as theme, workspace tab display, and docked window options. For more information about this dialog box see "Design Environment Options Dialog Box ".

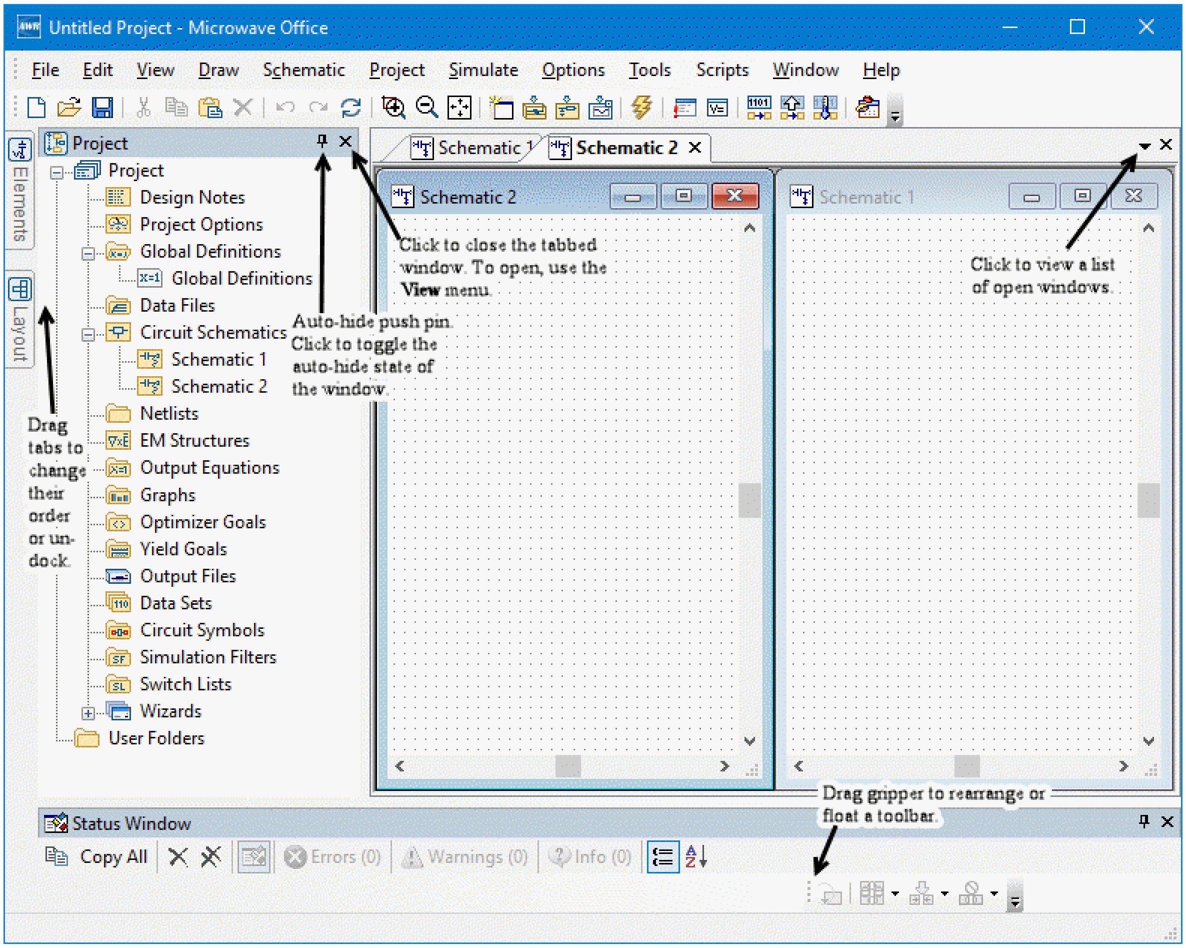


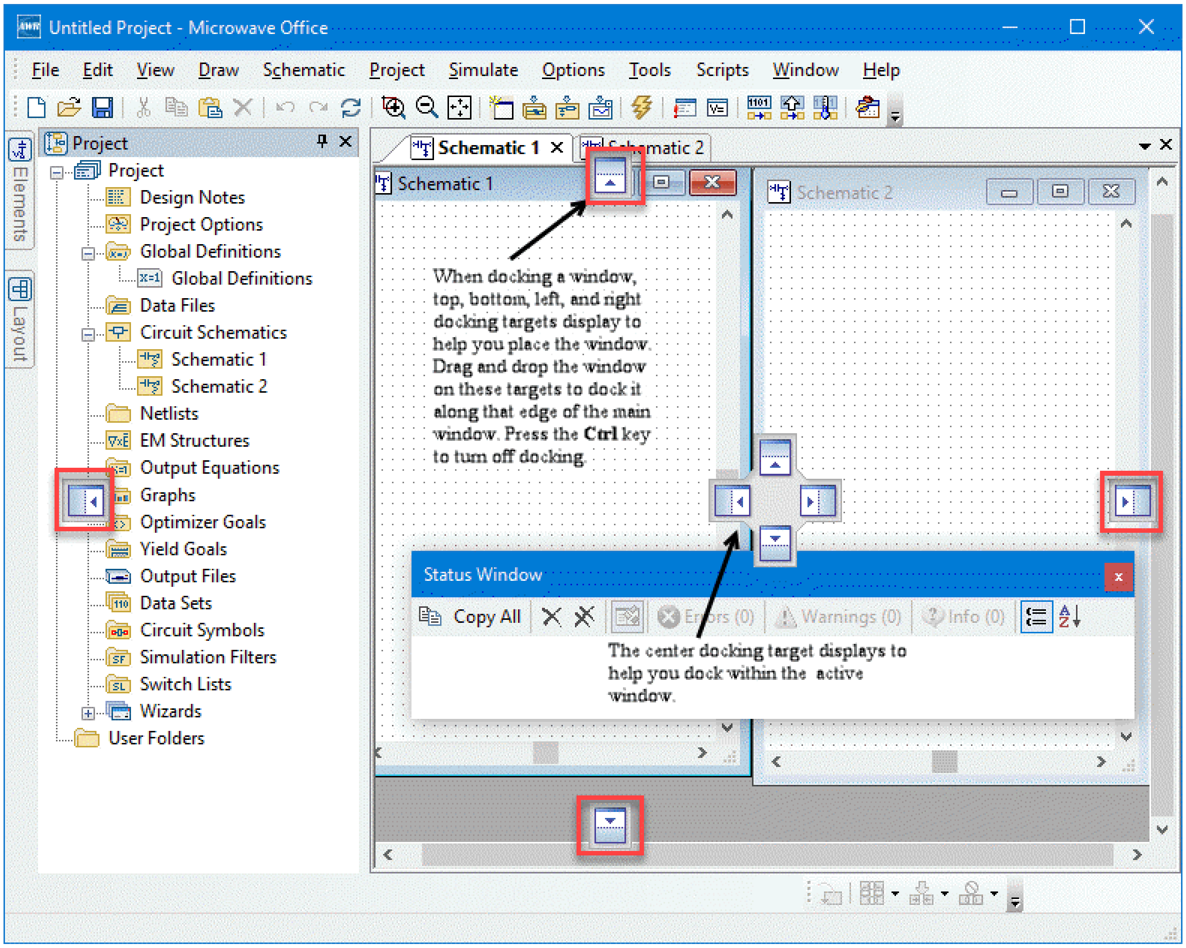
2.5.1.1. Docking Workspace Windows and Toolbars

The Project, Elements, Layout, and Status windows can be *docked* (displayed as tabs along the main window frame) or *floating* (fully displayed within the workspace). While docked, you can place these windows in auto-hide mode by clicking the "push pin" (Auto Hide toggle) icon in the upper right corner of the window. Auto-hiding windows disappear from view (and resume as tabs) shortly after you click elsewhere on the screen. By clicking the header of an unhidden window and dragging the window, you can dock the window on any edge of the main window, or float it anywhere in the workspace. During the move, a docking target displays to assist you in placing the window in the desired location. While dragging, position your cursor over the docking target arrow of the desired orientation, wait for the screen to highlight in that area and then release the mouse button to dock. To display a hidden window, click on or hover the mouse cursor over that window's tab. You can also access hidden windows using the associated **View** menu commands.

To close a docked or floating window click the "x" (Close) icon in the upper right corner of the window. To reopen a closed window, choose **View** and the appropriate window option.

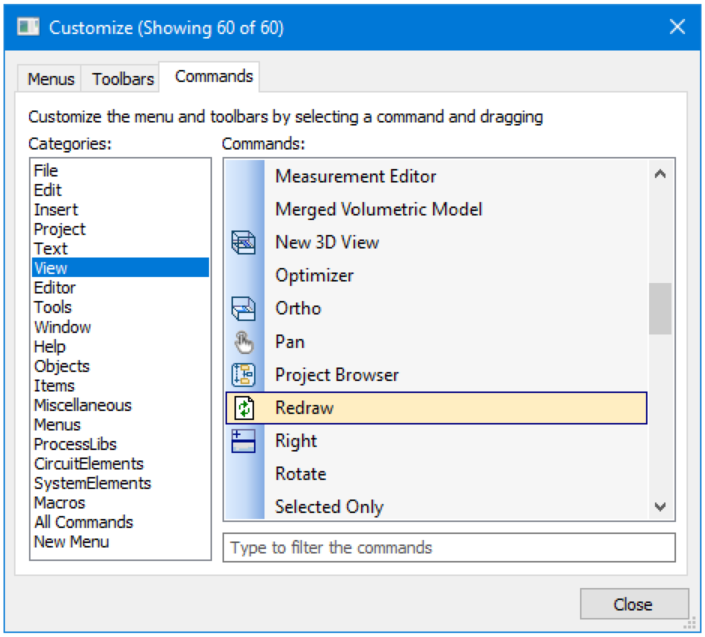
You can also float (fully display in the workspace) or dock (display inline along any edge of the main window frame) toolbars. To re-dock a floating toolbar, just double-click the toolbar title bar. To dock a toolbar in other than its default (top of the workspace) location, click on the dotted gripper that displays at the left side of the toolbar and drag the toolbar to another edge of the main window until it aligns with the frame, then release the mouse button.


Untitled Project - Microwave Office
File Edit View Draw Schematic Project Simulate Options Tools Scripts Window Help
Project
Elements
Layout
Project
Design Notes
Project Options
Global Definitions
Global Definitions
Data Files
Circuit Schematics
Schematic 1
Schematic 2
Netlists
EM Structures
Output Equations
Graphs
Optimizer Goals
Yield Goals
Output Files
Data Sets
Circuit Symbols
Simulation Filters
Switch Lists
Wizards
User Folders
Drag tabs to change their order or un-dock.
Auto-hide push pin. Click to toggle the auto-hide state of the window.
Click to close the tabbed window. To open, use the View menu.
Schematic 1
Schematic 2
Click to view a list of open windows.
Drag gripper to rearrange or float a toolbar.
Status Window
Copy All
Errors (0)
Warnings (0)
Info (0)



2.5.2. Customizing Toolbars and Menus

The Customize dialog box (**Tools > Customize**) has tabs for customizing and configuring the content of toolbars and menus in the AWR Design Environment platform. Note that all customizations apply only to the current menu and toolbar, they are not globally applied. The Customize dialog box must be open for the following customization steps to work.



2.5.2.1. Customize Dialog Box: Menus Tab

If not currently displayed in the program, to display a menu you want to customize, click the **Menus** tab in the Customize dialog box and select the desired menu. The menu set displayed in the menu bar at the top of the AWR Design Environment platform workspace area changes to reflect your selection. Click the **Reset** button to restore the selected menu to its defaults and remove any changes. With the Customize dialog box open, you can right-click on any menu name in the menu bar to display a list of customization options for that menu. To customize options within a menu, display the menu and then right-click on a menu option:

- **Reset**: Resets all changes made to the menu/option.
- **Delete**: Removes the menu/option from the menu/bar.
- **Name**: Allows you to change the default menu/option name. To create menu/option hotkeys, precede the hotkey (underlined) letter with an ampersand. For example, to access the **Project** menu using its hotkey **Alt + P**, the menu is named "&Project".
- **Copy Button Image**: Allows you to copy the corresponding toolbar button image for use in a program that supports this operation. Not all menu options have corresponding toolbar button images.
- **Paste Button Image**: Allows you to replace the default corresponding toolbar button image with an image copied from a program that supports this operation. Not all menu options have corresponding toolbar button images.
- **Reset Button Image**: Resets a revised button image to its default image.
- **Edit Button Image**: Opens a simple button editor to allow you to make changes to the current corresponding button image.
- **Change Button Image**: Provides a number of images from which you can choose to replace the current corresponding button image.
- **Default Style**: For menu items with corresponding button images, displays both the menu/option name and image on the toolbar.
- **Text Only**: Displays the menu/option name only (no image) on the toolbar.

- **Image and Text**: For menu items with corresponding button images, displays both the menu/option name and image on the toolbar.
- **Begin a Group**: Places a divider bar to the left of the menu on the menu bar or above an option in a menu for customized menu/option group organization.

2.5.2.2. Customize Dialog Box: Toolbars Tab

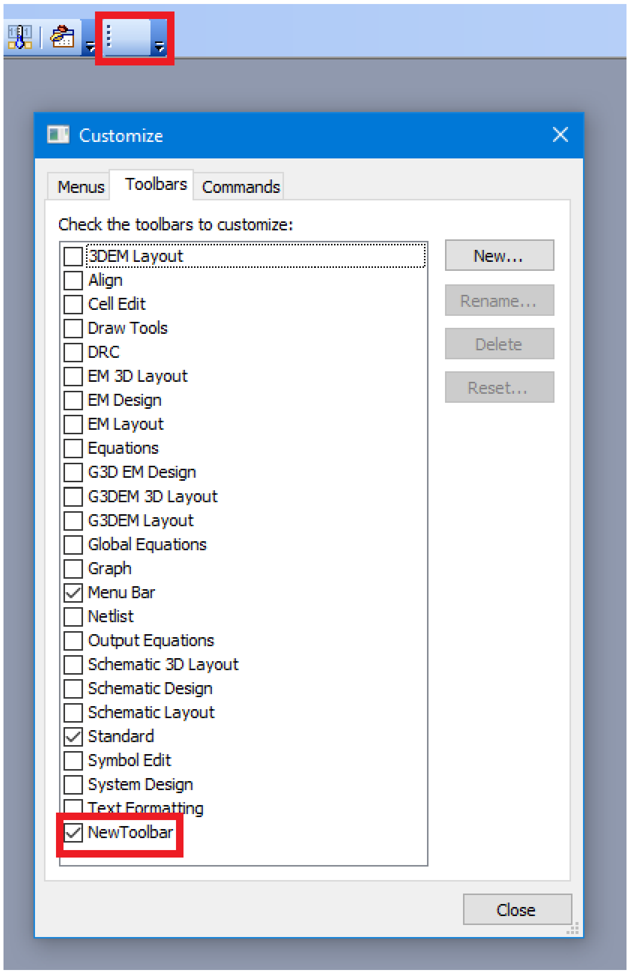
If not currently displayed in the program, to display a toolbar you want to customize, click the **Toolbars** tab in the Customize dialog box and select the desired toolbar. Click the **New** button to create and name a new toolbar. Click the **Reset** button to restore a selected toolbar to its defaults and remove all changes. With the Customize dialog box closed, you can also view this list of toolbars by right-clicking anywhere on a toolbar in the main window. With the Customize dialog box open, you can right-click on any toolbar icon to display a list of customization options for that command button:

- **Reset**: Resets all changes made to the command button.
- **Delete**: Removes the command button from the toolbar.
- **Name**: Allows you to change the default button name.
- **Copy Button Image**: Allows you to copy the button image for use in a program that supports this operation.
- **Paste Button Image**: Allows you to replace the default button with an image copied from a program that supports this operation.
- **Reset Button Image**: Resets a revised button image to its default image.
- **Edit Button Image**: Opens a simple button editor to allow you to make changes to the current button image.
- **Change Button Image**: Provides a number of images you can choose to replace the current button image.
- **Default Style**: Displays the command button as an image on the toolbar.
- **Text Only**: Displays the command button name only (no image) on the toolbar.
- **Image and Text**: Displays both the command button name and image on the toolbar.
- **Begin a Group**: Places a divider bar to the left of the button on the toolbar for customized button group organization.

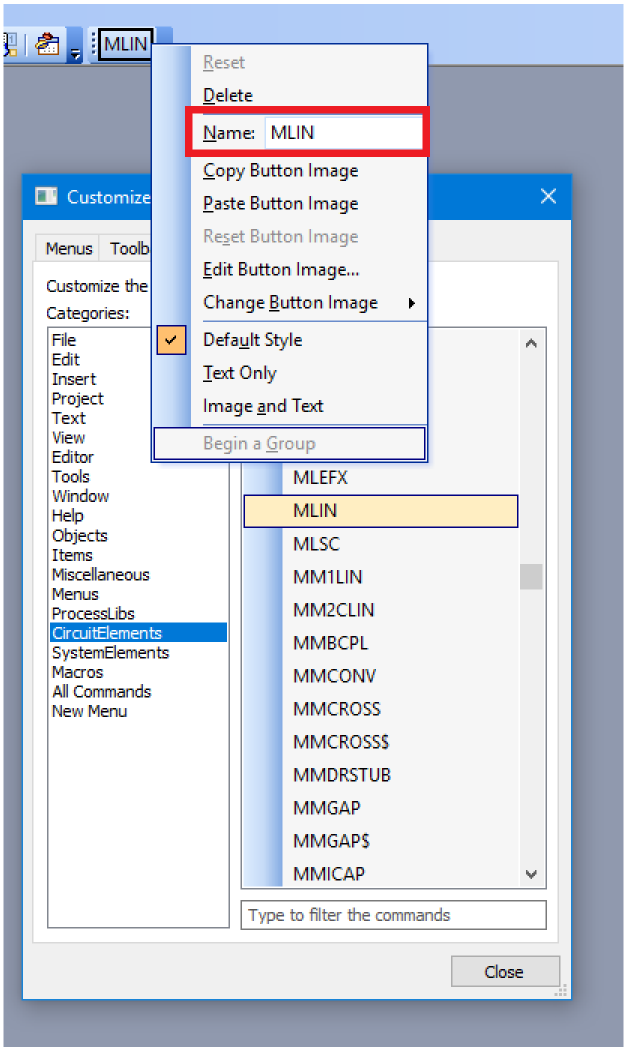
Adding a Custom Toolbar and Button

The following example shows how to add a new toolbar with a random customized button. You can add multiple buttons to new or existing toolbars using these steps.

1. Choose **Tools > Customize**, and on the Customize dialog box **Toolbars** tab, click the **New** button.
2. In the New Toolbar dialog box, type a name for your new toolbar and click **OK**. The new toolbar name displays at the bottom of the toolbar list in the dialog box, and an empty toolbar is created on the toolbar in the main window.



3. On the Customize dialog box **Commands** tab, under **Categories**, select "CircuitElements" and then under **Commands** search for and select the MLIN element.
4. Drag the MLIN element to the new empty toolbar at the top of the main window, and drop it.
5. Right-click the new MLIN button for options to change its name or other characteristics, including using a stock image (**Change Button Image**) or opening the Button Editor to create a custom image (**Edit Button Image**).



6. Click **Close** to save your toolbar and close the Customize dialog box.

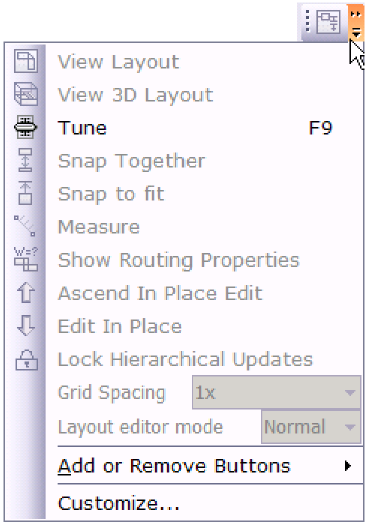
2.5.2.3. Customize Dialog Box: Commands Tab

The Customize dialog box **Commands** tab contains groups of categorized commands as well as elements, libraries, and macros that you can drag to menus or toolbars to customize their content.

To add a command to a displayed menu set, select the command category in **Categories**, then click on the desired command in the associated **Commands** list. Drag the command to the menu of choice and drop it to add it. You can add commands directly to the menu bar or hold the mouse over a menu while dragging to display that menu and add the command as an option. To view all available commands, select **All Commands**.

To add a command to a toolbar, select the command category in **Categories**, then click on the desired command in the associated **Commands** list. Drag the command to the visible toolbar of choice and drop it to add it.

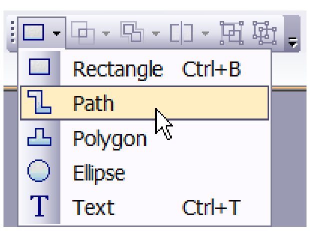
When the combined width of all docked toolbars exceeds the width of the main window frame, some toolbars are "compressed" and a chevron button displays at the end of the toolbar. Click this button and choose **Customize** as an alternate way to display the Customize dialog box, or choose **Add or Remove Buttons** to view a list of buttons/commands included in the toolbar. Select or clear the check boxes for the individual commands to include or remove them from the toolbar. Click **Reset Toolbar** at the bottom of this list to restore the toolbar to its default command list.



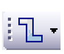
Split Buttons

Split buttons combine a single command with an arrow you can click to access other similar commands in a menu format. A split button consolidates commands and saves space on the toolbar, while remembering and displaying the last command you used from the group. For example, on the Draw Tools toolbar, clicking the rectangle icon at the left of the following split button allows you to draw a rectangle.

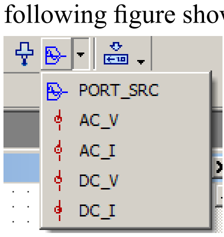
Clicking the down arrow at the right of this split button displays a drop-down menu of all commands associated with the button.



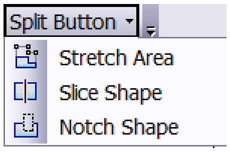
In this example, if you choose a command other than **Rectangle**, the icon for that command replaces the rectangle icon on the button face when the drop-down menu closes.



You can edit existing split buttons or add split buttons to any toolbar to make your own groups of commands. The following figure shows the split button on the Schematic Design toolbar that combines all the dynamic sources.



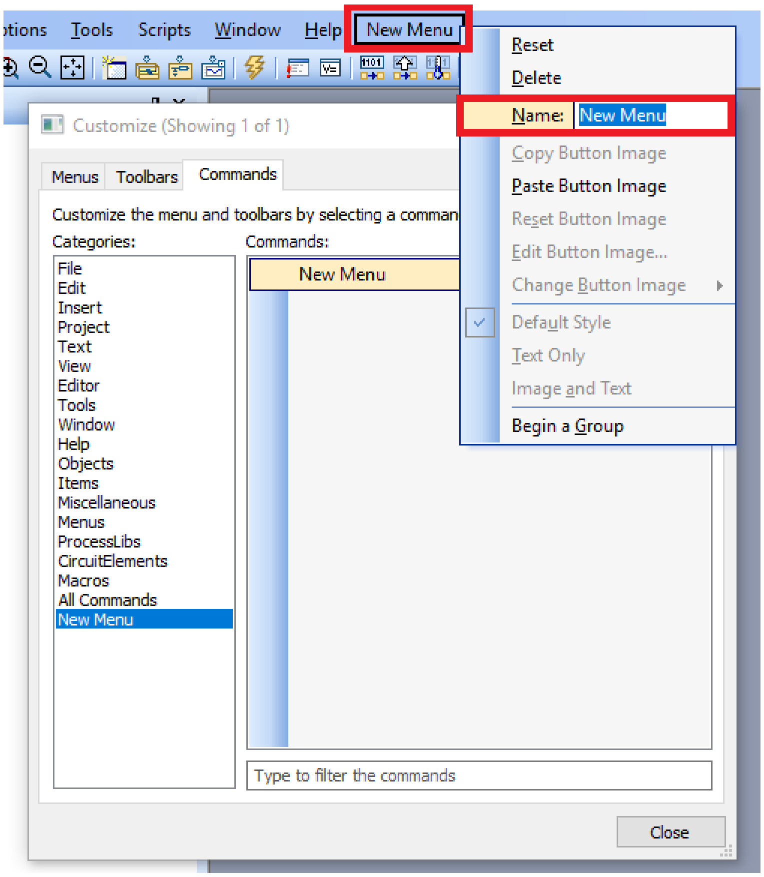
To create a split button, choose **Tools > Customize**. In the Customize dialog box, click the **Commands** tab and then select **Menus** in the list of **Categories**. Select **Split Button** under **Commands**, and then drag this item to the desired toolbar and drop it. Click the new button to display a blank drop-down menu, and then drag and drop **Commands** from any of the **Categories** onto this menu to add them to the group. The following figure shows three commands added to the menu of a new split button.



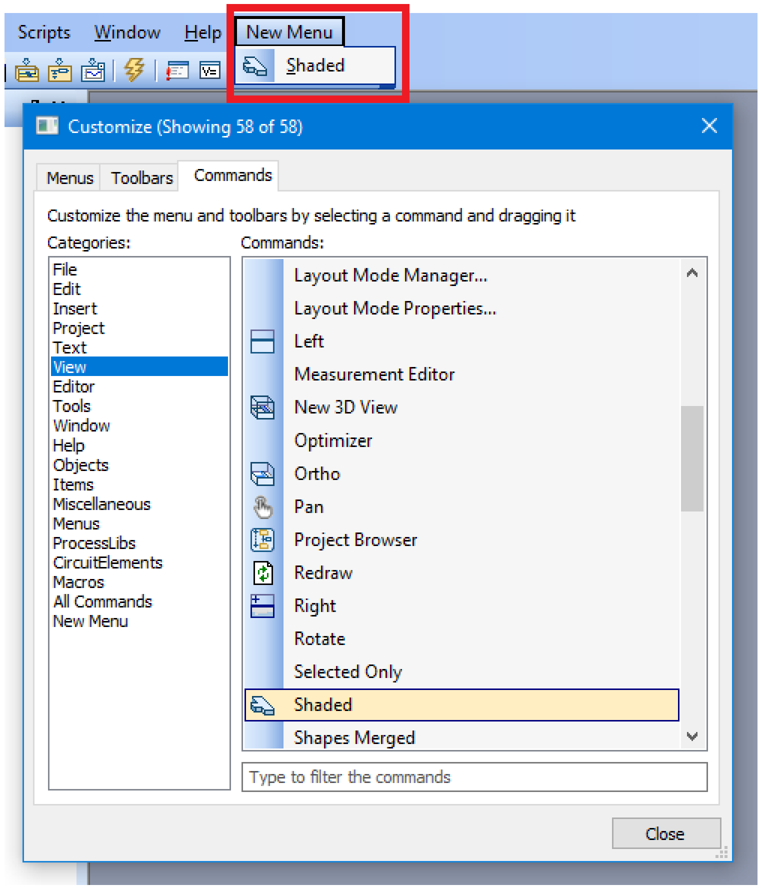
Adding a Custom Menu and Command

The following example shows how to add a new menu with customized commands. You can add multiple commands to new or existing menus using these steps.

1. Choose **Tools > Customize**, and on the Customize dialog box **Commands** tab, under **Categories**, select "New Menu".
2. Under **Commands**, drag the "New Menu" to the menu bar at the top of the main window, and drop it.
3. Right-click the new menu to change its name, then press the **Enter** key.



4. In the Customize dialog box, add commands to the menu by choosing a category under **Categories**, and a command under **Commands**, and then dragging and dropping the commands onto the new menu.

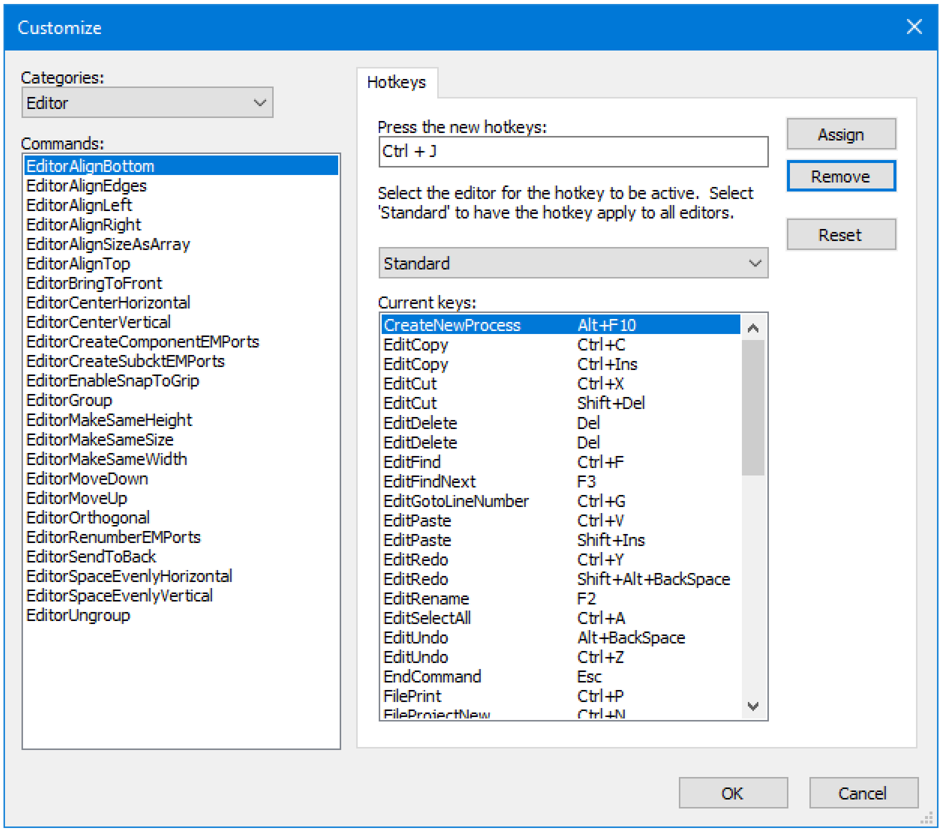


5. Click **Close** to save your menu and close the Customize dialog box.

2.5.3. Assigning and Configuring Hotkeys

The Customize dialog box (**Tools > Hotkeys**) allows you to assign and configure hotkeys for a wide variety of program elements and commands. **NOTE**: Menu command shortcuts override hotkey assignments.

Select a category in **Categories** to display associated commands. Select a command and then type the desired hotkey or key combination at the top of the Hotkeys tab. Apply the hotkey(s) to an editor in the drop-down list, or choose **Standard** to apply the hotkey universally, then click **Apply**. You can change a default hotkey assignment by selecting it in the **Current keys** list and typing an alternate hotkey, and you can remove an assignment from this list by clicking **Remove**. To reset a command to its default hotkey, select it and click **Reset**.



2.5.4. Script Utilities

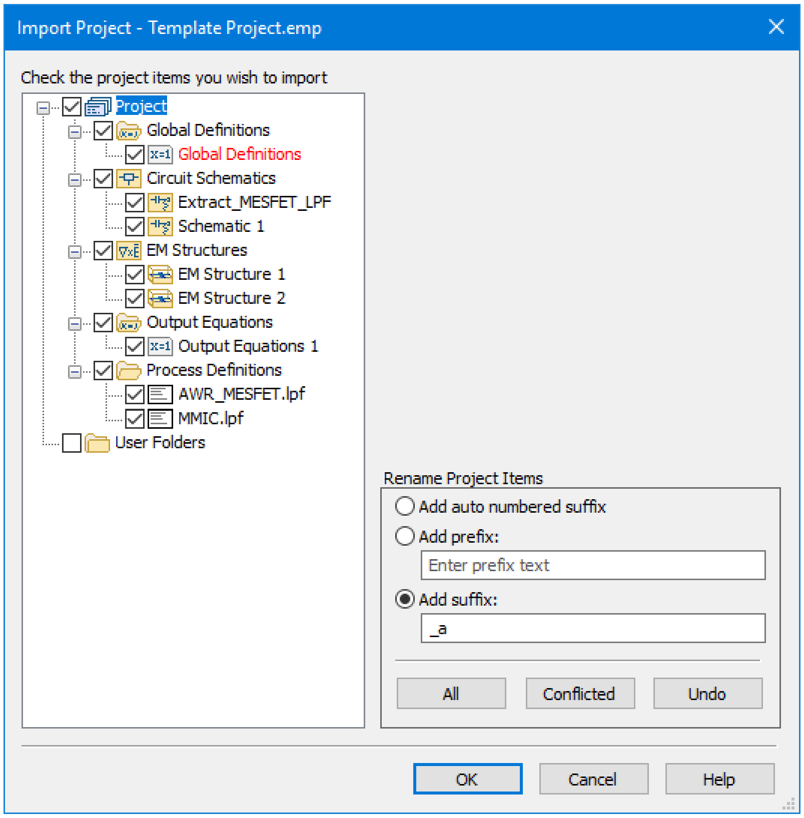
The AWR Design Environment platform includes many useful user utilities accessible on the **Scripts** menu. You do not need to know a scripting language or development environment to use the scripts.

Scripting is a great way to automate repetitive tasks. If you are interested in scripting, Cadence provides a scripting development environment, a full description of the API objects available, and many examples of how to use each object. For more information, see the AWR API Scripting Guide.

2.6. Importing a Project

You can import project items such as schematics, system diagrams, netlists, EM structures, graphs, Switch Lists, symbols, and others into the current (host) project.

To import another project into the current project, choose **File > Import Project**, then browse for the target project. As shown in the following figure, after you select the project, the Import Project dialog box displays a project tree to allow you to select the project items you want to import.



Items that display in red are *conflicted* (have the same name in the host project). For conflicted items, you can perform one of the following actions:

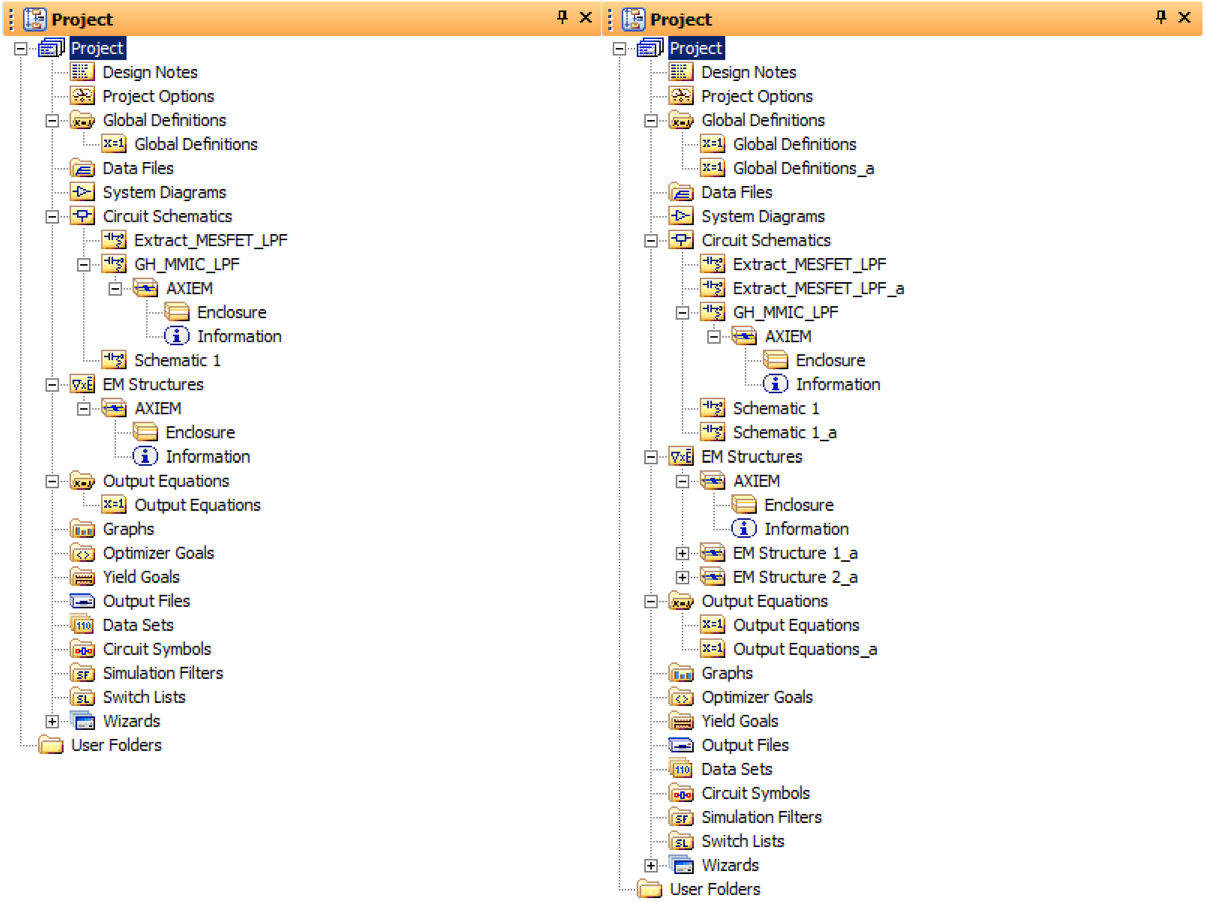
- Select **Add auto numbered suffix** to allow the AWR Design Environment software to rename the selected, conflicted items by adding a number as a suffix. You must click the **Conflicted** button to rename the items.
- Select **Add prefix** to rename the items by adding a prefix you specify. The prefix can only be A-Z, a-z, 0-9, or the "_" or " " (blank) characters. You can add a prefix to just the selected, conflicted items by clicking the **Conflicted** button, or add a prefix to all selected items in the project tree by clicking the **All** button.
- Select **Add suffix** to rename the items by adding a suffix you specify. The suffix can only be A-Z, a-z, 0-9 or the "_" or " " (blank) characters. You can add a suffix to just the selected, conflicted items by clicking the **Conflicted** button, or add a suffix to all selected items in the project tree by clicking the **All** button.
- Manually rename each item by right-clicking it in the project tree and choosing **Rename**.
- Click **OK** to import all of the items and overwrite the conflicted items.
- Click **Undo** to revert all **Rename Project Items** operations performed since opening the dialog box.

NOTES:

- Child folder items mirror the state of their parent folder items (either selected or cleared).
- If there are no child folder items (system diagrams in this example), the parent folder (**System Diagram**) is hidden.
- Items under **User Folders** track the state of the corresponding items under the **Project** node (either selected or cleared).
- Renaming items during project import updates references within the imported project only; it does not affect the host project or the import project disk image.

- You cannot import Design Notes, project scripts, and Wizard instance data.

The following figure shows the project tree before (left) and after (right) the import specified in the previous figure.



2.6.1. Host and Import Project Differences

When there are differences between the host project and the import project, the following standards are in effect upon project import:

- If the host project and the import project have different versions of the same PDK, the host project PDK version is used.
- If the host project and the import project use different units, a warning displays for any schematics that use variables with a link, so you can locate them. Models with intelligent syntax such as W@1 typically do not prompt warnings because they get the correct values from elements connected to them.
- Differences between the import project and the host project in the following layout options may cause the layout to differ after import:
  - **Database unit size** (choose **Options > Layout Options > Layout** tab)
  - **Auto face inset** (choose **Options > Layout Options > Layout** tab)
  - **Number of points/circle** (choose **Options > Layout Options > Layout** tab)
  - **Fixed origin for subcircuits** (choose **Options > Layout Options > Layout** tab)

- **Fixed origin for layout cells** (choose **Options > Layout Options > Layout** tab)
- **Allow pCell's origin to float** (choose **Options > Layout Options > Layout** tab)
- **Draw as polygons** (choose **Options > Layout Options > Paths** tab)
- **Default Model** (choose **Options > Layout Options > iNet** tab)

2.7. Archiving a Project

You can archive into one file your project and any external files on which it depends. Archiving is useful for storing projects or for sending projects to others for review or support. When an archive is reopened, its separate files are reconstituted and links from the project to each file are re-established.

The following items are stored in a project archive in addition to the standard objects such as graphs, schematics, and system diagrams:

- PDKs loaded in the project
- Linked data files, schematics, system diagrams, EM structures, and GDS/DXF libraries
- Global user scripts
- Data sets (even if your project is not set to save data sets in it)
- Extended multipaths

The following items are **not** stored in a project archive:

- Files under the *Signals* subdirectory
- Files that contain symbolic directories such as "$PRJ" in their paths.
- Netlist library (*.lib*) files referenced in HSPICE/Spectre netlists
- Files referenced as hard-coded in LabVIEW models
- Files referenced by scripts

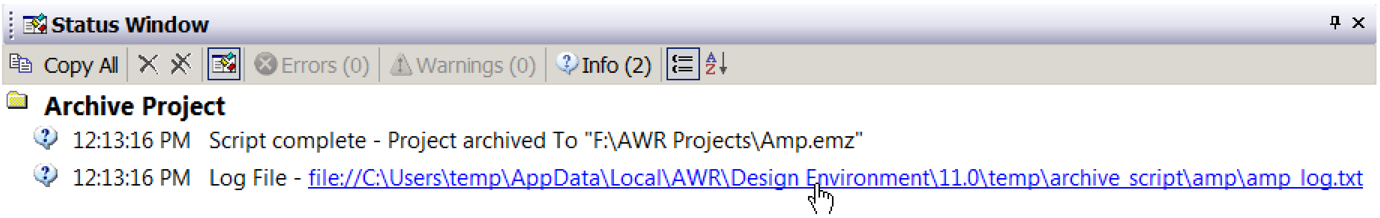
To archive a project, you must first save it by choosing **File > Archive Project**, or **File > Save Project As** and manually changing **Save as type** to **Project Archive (*.emz)**.

By default the archive has the same name as the project, but with an *.emz* extension.

There are a few convenient ways of archiving the project for different purposes:

- To send the archived project by email, choose **File > Email Project > Project and Dependencies**.
- To contact Cadence AWR Support for help, choose **Help > Get Technical Support**.

Archiving a project can take time, depending on how big or complicated the project is. When the archive is complete and the *.emz* file is created, the Status Window displays with information about the archive process. One of the items in the Status Window is a link to the log file, which includes a detailed list of the project contents, and the external files referenced by the project.



Opening an archived project is similar to opening a regular project; just choose **File > Open Project**, but set the file type to **Project Archive (*.emz)**. When the archive opens, a folder with the same name and location as the project is created, and contains the external files.

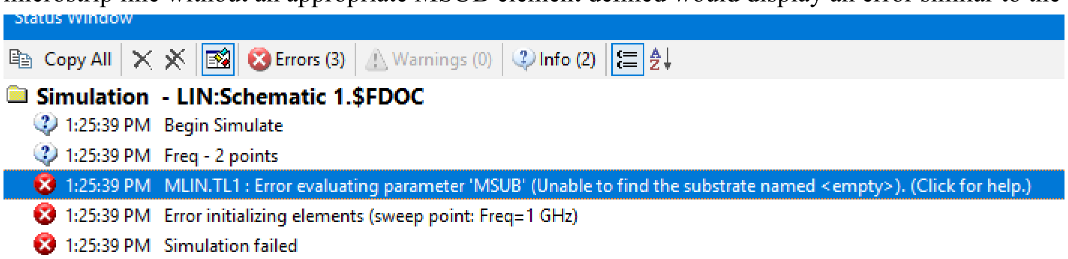
NOTE: You cannot rename Project Archive (*.*emz*) files once they are created; they will fail to open.

2.8. Viewing the Status Window

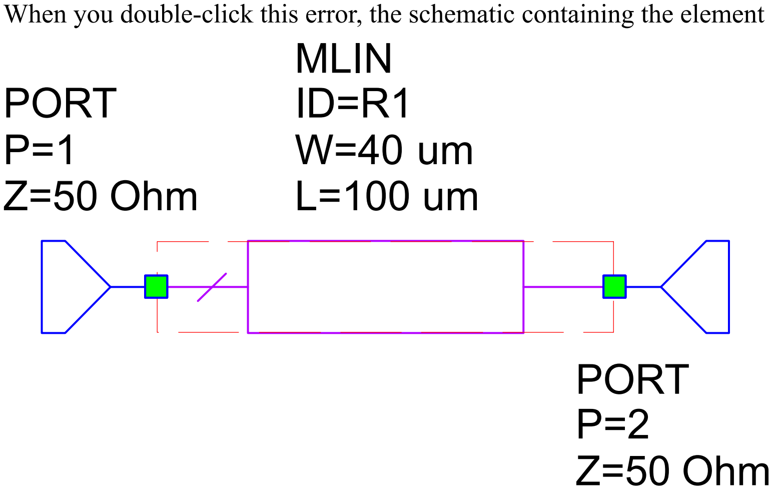
The Status Window (located at the bottom of the main window when docked) displays three different types of messages issued from various operations:

- Errors: These messages are errors in the simulation or the project.
- Warnings: These messages are warnings in the simulation or the project.
- Info: These messages provide information about the simulation status and the project.

You can double-click certain message types in the Status Window to navigate to the message source. For example, a microstrip line without an appropriate MSUB element defined would display an error similar to the following.

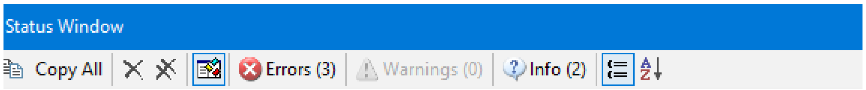


When you double-click this error, the schematic containing the element displays with the MLIN highlighted in red.



2.8.1. Status Window Controls

The Status Window includes a toolbar as shown in the following figure:

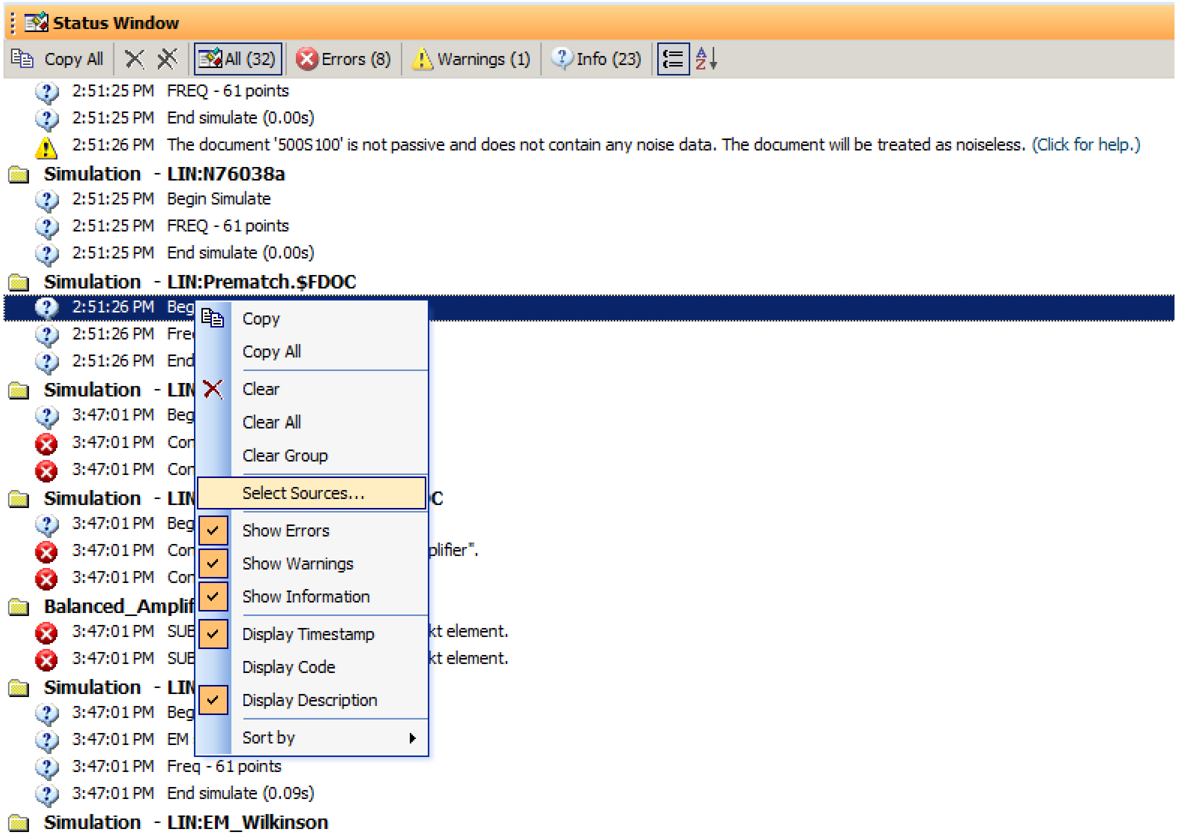


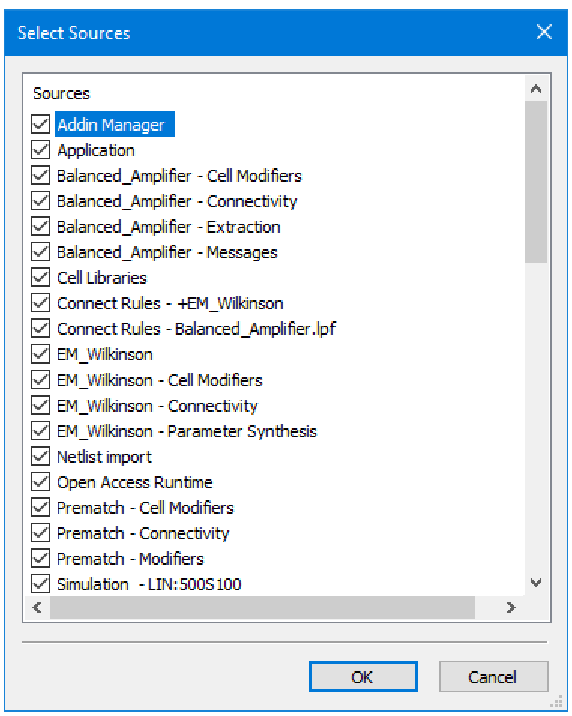
The Status Window toolbar commands include:

- **Copy**: Copy the selected item to the Clipboard.
- **Copy All**: Copy all of the items in the Status Window to the Clipboard.
- **Clear**: Clear the selected item from the Status Window.
- **Clear All**: Clear all of the items from the Status Window.
- **All**: Display all of the items from the different message categories (errors, warnings and info).
- **Errors**: Display only error items.
- **Warnings**: Display only warning items.
- **Info**: Display only simulation information items.
- **Show in Groups**: Group items into their sources (for example, schematics, system diagrams, and EM structures).
- **Sort**: Sorts the items based on either time stamp, type, code, or description. **NOTE:** Code is reserved for future use.

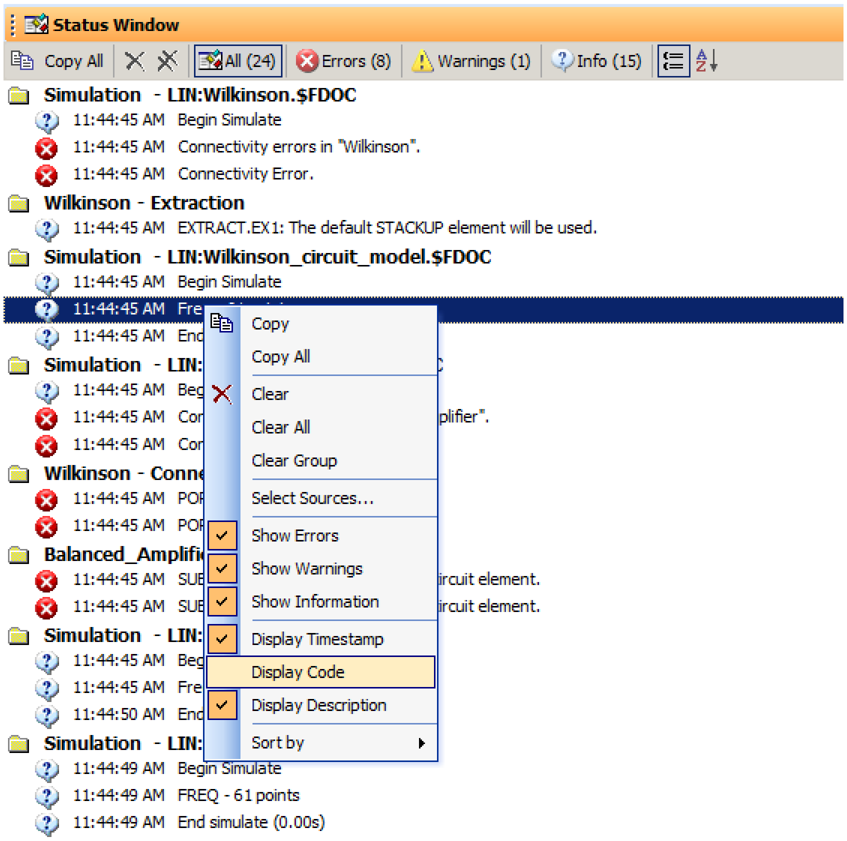
You can right-click an error to access these and additional commands, as shown in the following figures.

You can select which items display in the Status Window based on the item source.

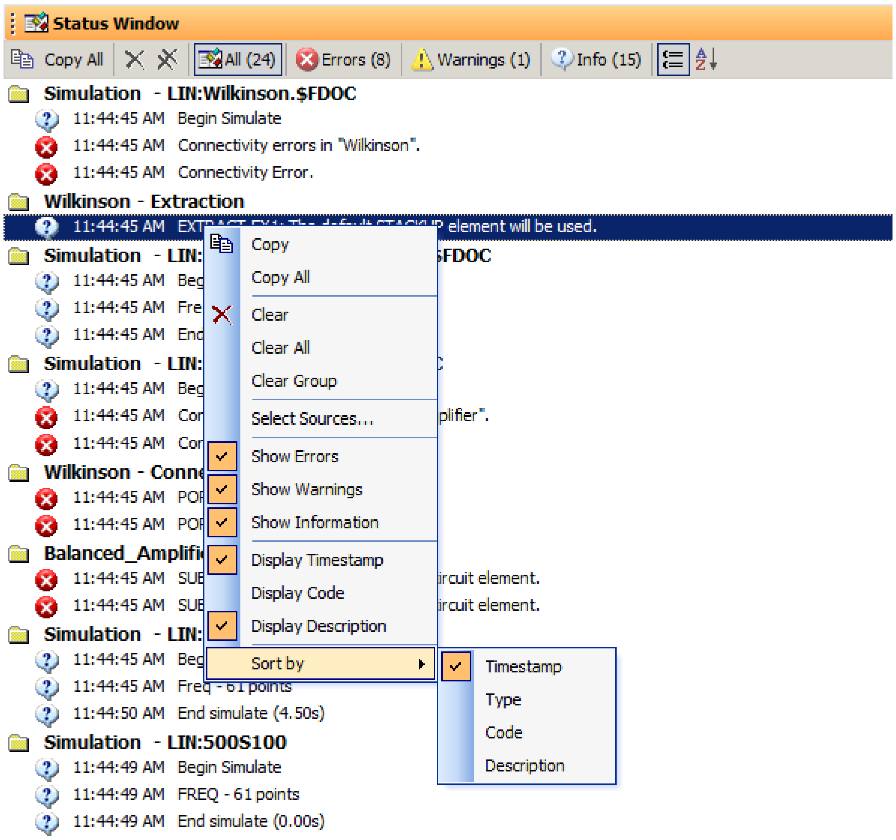




You can specify whether or not the timestamp, code, or description of each item displays.



You can then sort items in the Status Window by timestamp, type, code, or description.



2.9. Using Version Control

Version control is the process with which documents/data file(s) are recorded and managed with revision control software that stores them in a central database or repository. Commands you enter instruct the software on how to manage the target file(s) in a repository, and the commands are added to the file history. Most version control software (VCS) supports descriptive comments when a command is executed, adding these details to the target file history. The software manages and allows access to the file history and database of all files stored within. For group design, version control software file management prevents unintentional file overwrites when there are multiple users editing the same file in the version control database or central repository.

The AWR Design Environment platform supports integration of Version Control software to effectively manage group design projects, allowing group design data management of complex, multi-function designs on many different technologies. The currently supported version control software vendors are Subversion and ClioSoft.

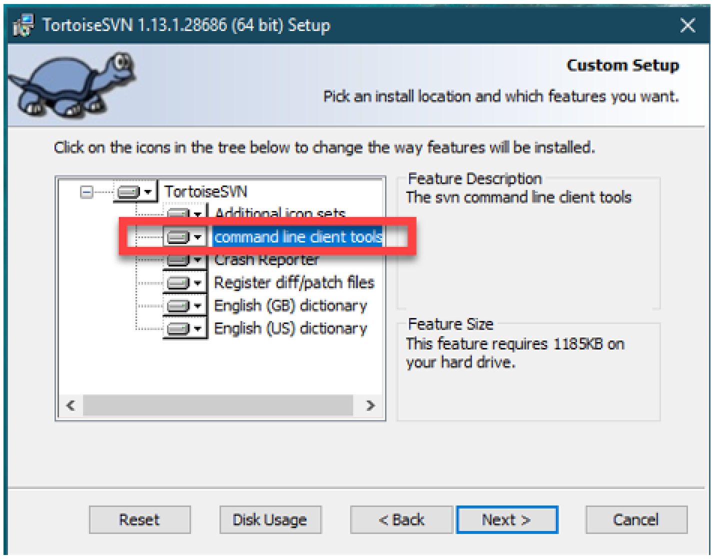
To utilize this feature, you must have a license for version control.

For more information about the version control user interface, see "Version Control Dialog Box".

2.9.1. Vendor-specific Setup

Ensure the following requirements are met for the Version Control software you use.

Subversion: Install the Command line Client option as shown in the following figure.



ClioSoft: Note these typical Windows Client environment variables. The license file variable assumes the license file is managed by FlexLM.

- SOS License Variable
  - Variable Name: CLIOLMD_LICENSE_FILE
  - Variable Value: port@host
- SOS Installation Dir Variable
  - Variable Name: CLIOSOFT_DIR
  - Example Variable Value: *C:\Program Files\ClioSoft\<ClioSoftVersion>*

DesignSync: These are DesignSync Windows Client environment variables. For licensing information, contact your DesignSync license administrator.

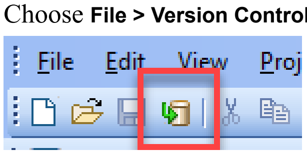
- DesignSync Installation Dir Variable
  - Variable Name: SYNC_DIR
  - Example Variable Value: *C:\Program Files\ENOVIA\Synchronicity\<DesignSyncVersion>*
- Add DesignSync bin directory to path variable
  - %SYNC_DIR%\bin

2.9.2. Connecting to a Repository

Follow the vendor-specific steps for connecting to an existing Version Control repository.

To connect to **Subversion**:

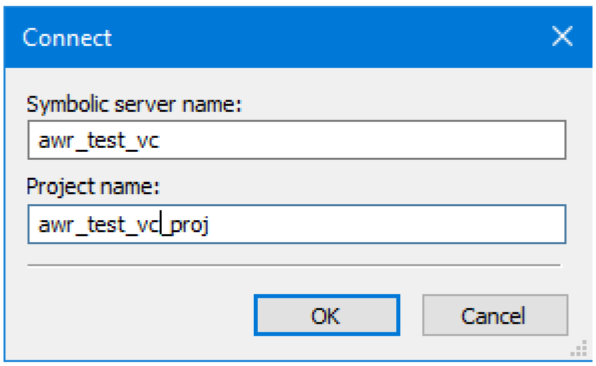
1. Choose **File > Version Control**, or click the **Version Control** button on the toolbar.



2. In the Version Control dialog box, click the **Connect** button on the toolbar to display the Connect dialog box.
3. Select **Subversion**, and then click **Connect**.
4. Enter the URL for your Subversion repository, then click **Download**.
5. Your Subversion repository displays under **Sources**.

To connect to **ClioSoft**:

1. Choose **File > Version Control**, or click the **Version Control** button on the toolbar.
2. In the Version Control dialog box, click the **Connect** button on the toolbar to display the Connect dialog box.
3. Select **ClioSoft**, and then click **Connect**.
4. Enter the existing **Symbolic server name** and **Project name** for your ClioSoft repository, then click **OK**.

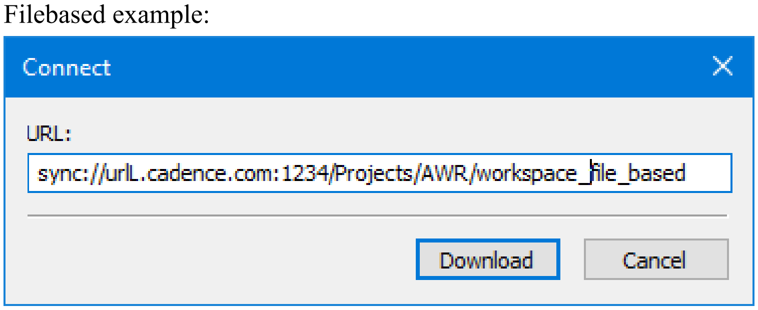


5. Your ClioSoft repository displays under **Sources**.

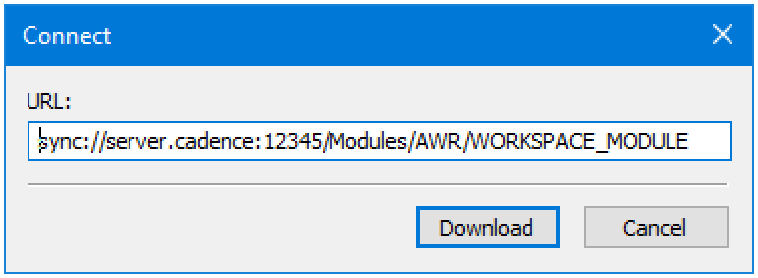
To connect to the **DesignSync Module** and **FileBased Vault**:

1. Choose **File > Version Control**, or click the **Version Control** button on the toolbar.
2. In the Version Control dialog box, click the **Connect** button on the toolbar to display the Connect dialog box.
3. Select **DesignSync**, and then click **Connect**.
4. Enter a server path for your FileBased or DesignSync Module, then click **Download**.

   Filebased example:



   Module example:



5. Your DesignSync repository displays under **Sources**.

2.9.3. Updating or Removing a Repository from AWR

To update or remove a repository, follow the steps for your repository type.

- To update a local repository folder in the **Sources** list
  1. Under **Sources**, select the repository for update.
  2. Click the **Update** button on the toolbar, or right-click the repository name and choose **Update**.
- To remotely update a repository in the **Sources** list.
  1. Under **Sources**, select the repository for update.
  2. Click the **Remote Status** button on the toolbar, or right-click the repository name and choose **Remote Status**.
- To remove a repository in the **Sources** list.
  1. Under **Sources**, select the repository you want to remove.
  2. Right-click and choose **Remove**, or click the **Remove** button on the toolbar.

2.9.4. Managing Documents from the AWR Version Control Dialog Box

Supported document types include:

- Global definitions
- Data files
- System diagrams
- Circuit schematics
- EM structures
- Netlists
- Circuit symbols
- Cell libraries (GDS and DXF)

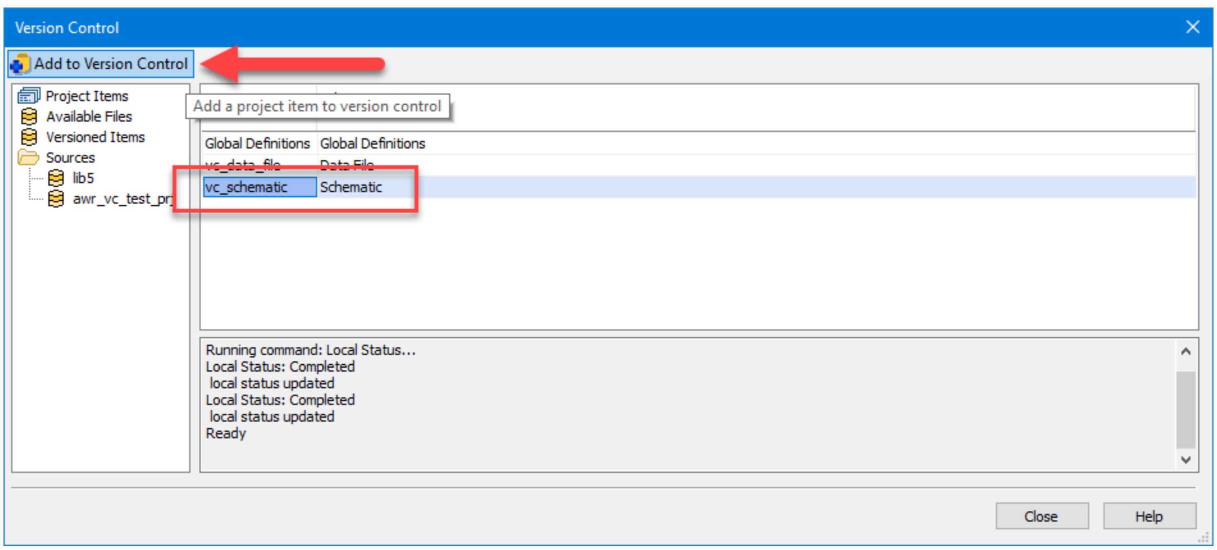
The following table shows local and remote status icons for various document states.

Local Status	VCS icon	Remote Status	VCS icon	Description
Adding	Adding	N/A	N/A	New local doc is ready for repository commit
Normal	Normal	Normal	Normal	Doc is not locked for local or repository editing

Local Status	VCS icon	Remote Status	VCS icon	Description
Normal	Normal	Normal (Lock)	Normal	Doc is locked in the repository and checked out by another user
Normal (Edit)	Normal	Normal (Lock)	Normal	Doc is locked locally for editing by the current user
Modified	Modified	Normal (Lock)	Normal	Doc is modified locally
Unversioned	Unversioned	N/A	N/A	Doc is not under version control management
Remove	Deleting	Normal	Normal	Doc is in removal state; still requires commit

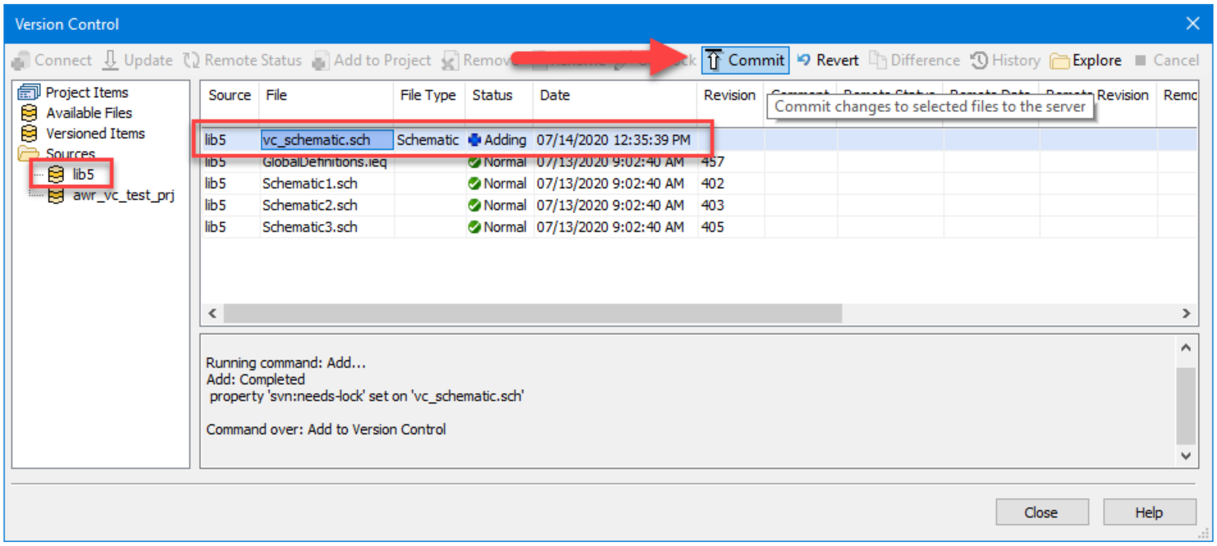
To add an unmanaged project document to a connected repository:

1. Select the documents/data files you want to add to the repository and click the **Add to Version Control** button on the toolbar. When adding/committing a hierarchical document, a second dialog box displays a list of dependent items that you can also select for addition to version control.

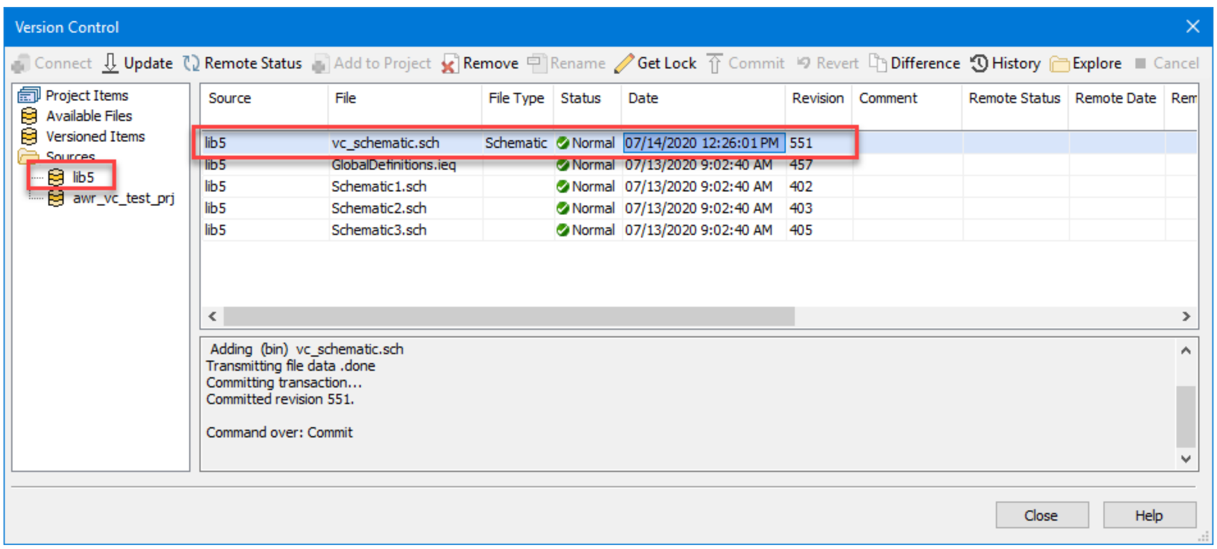


2. For **Subversion**, the document/data file displays in the repository with an "Adding" **Status**.

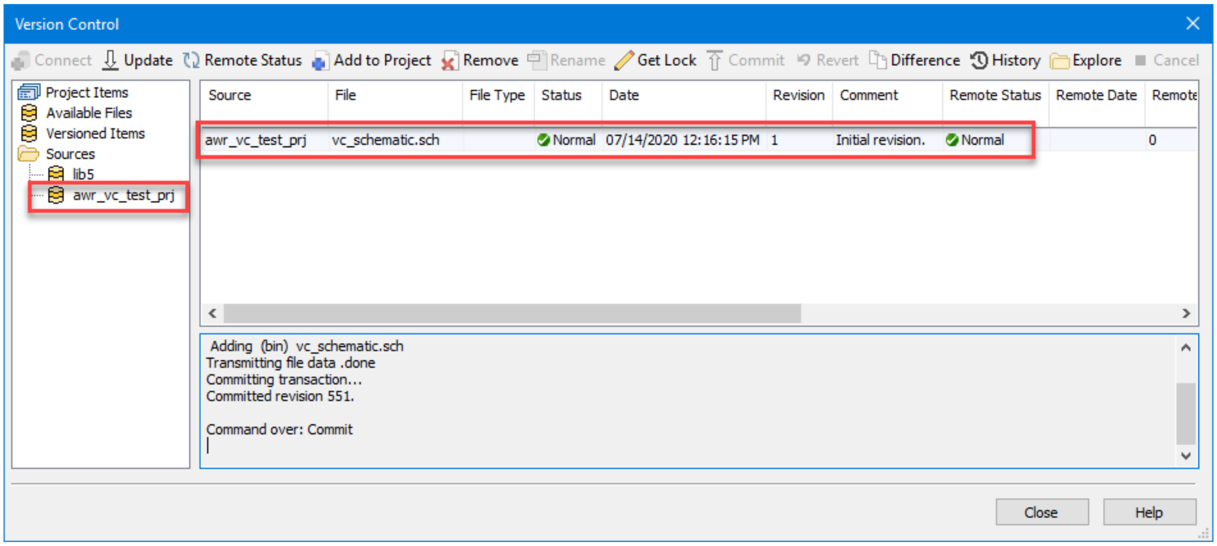
   To commit the document/data file, select it and click the **Commit** button on the toolbar.



The status displays as "Normal".

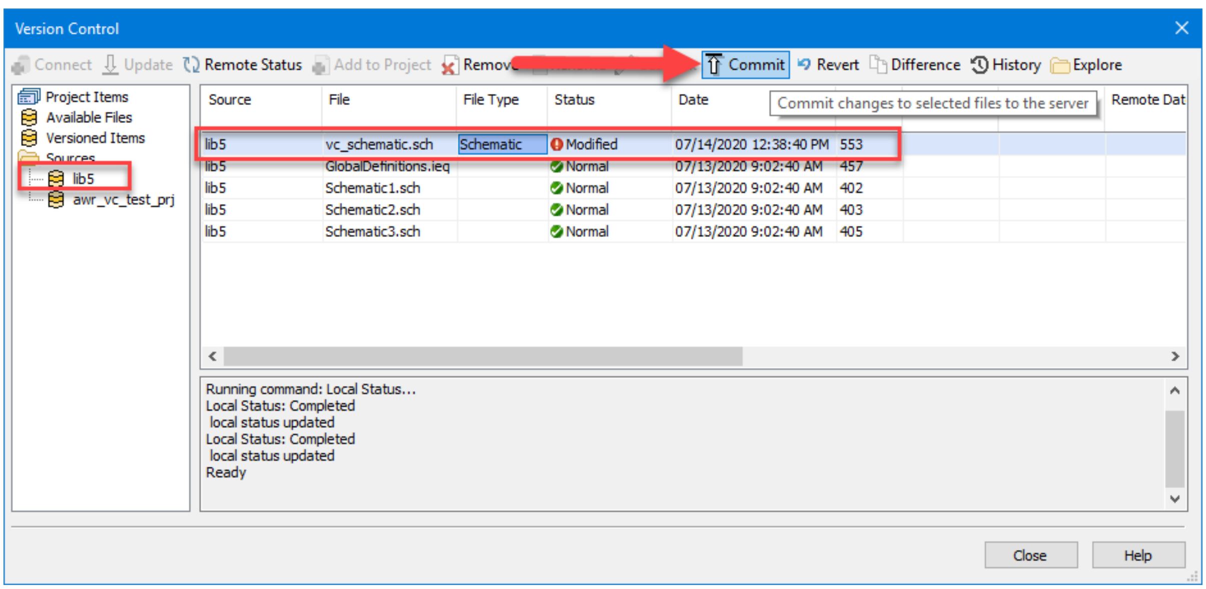


3. For **ClioSoft**, the document/data file displays in the repository with a "Normal" **Status**.



To commit a document to a connected repository:

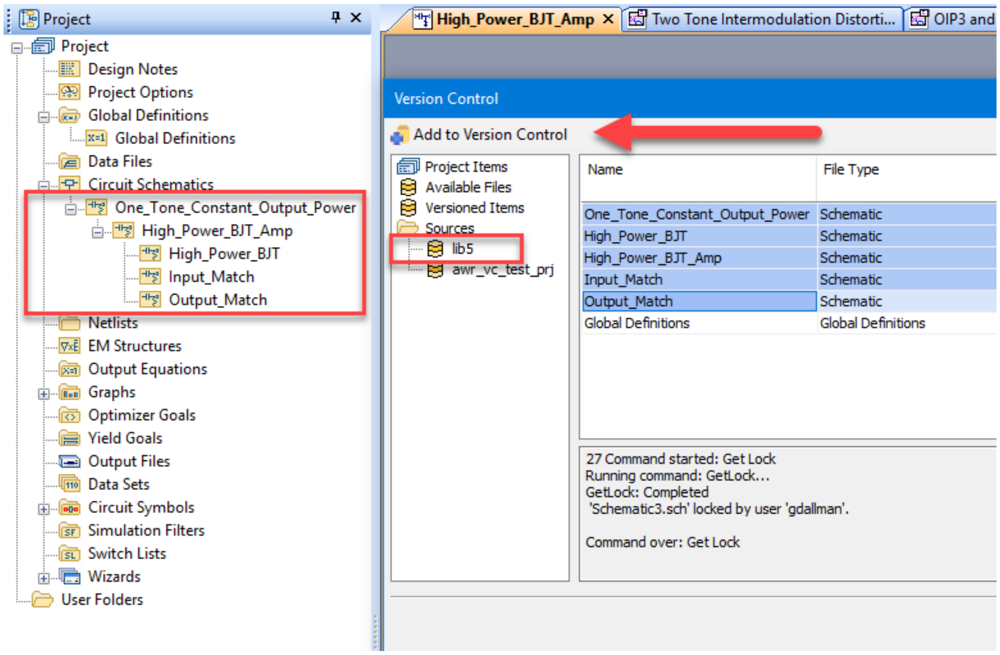
1. Select a changed document or (**Subversion** only) a newly added document and then click the **Commit** button on the toolbar.



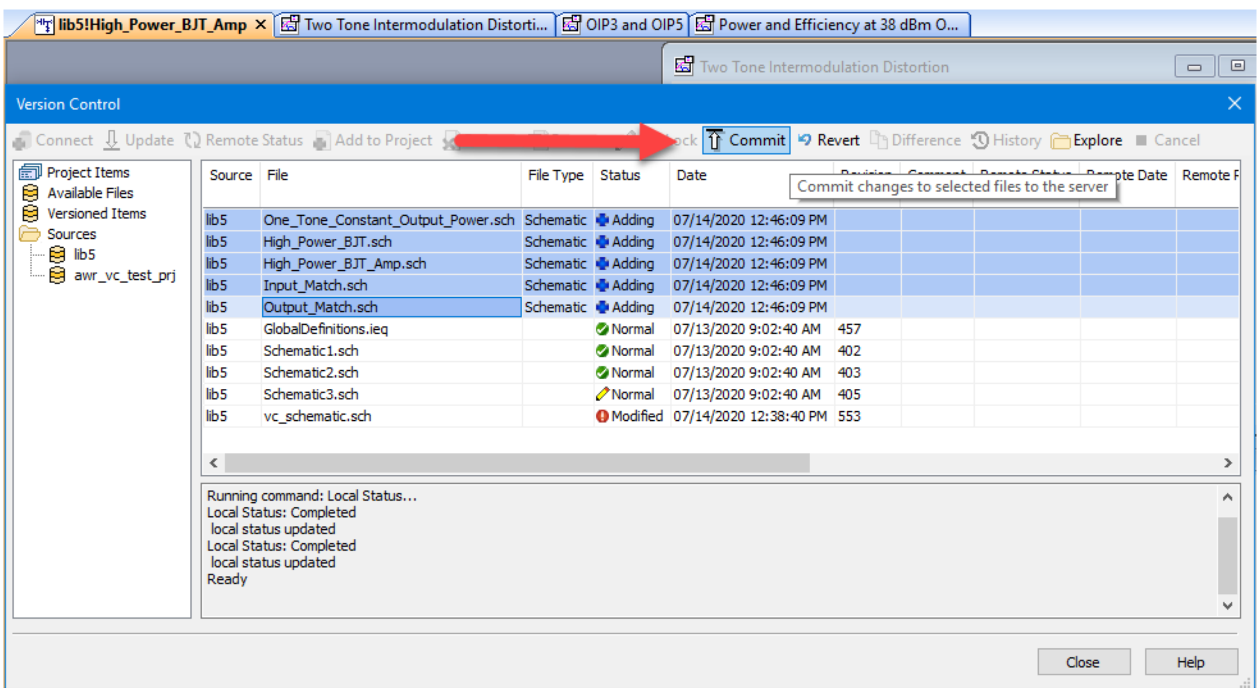
2. Add a minimum 5-character comment when prompted, then click **OK**.
3. The document/data file displays with a "Normal" **Status** and green check-mark.

To add and commit a hierarchical schematic:

1. In the Project Browser, select the parent schematic and all subcircuit schematics, then click the **Add to Version Control** button on the toolbar. The documents display in the repository.



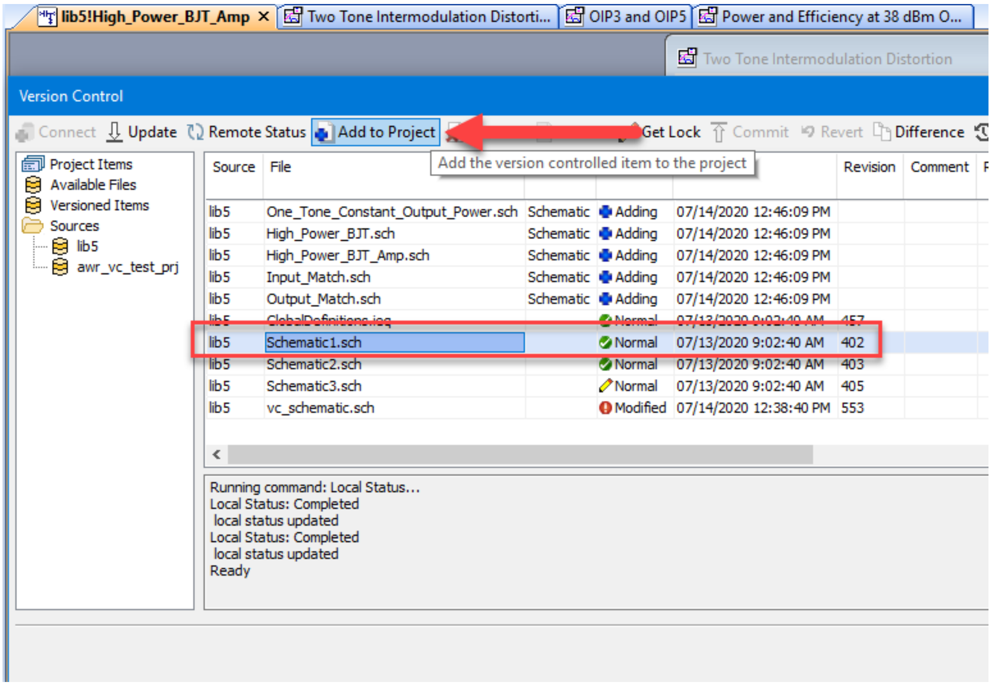
2. Select the documents and click the **Commit** button on the toolbar to commit them.



3. Add a minimum 5-character comment when prompted, then click **OK**.
4. Each document displays with a "Normal" **Status** and a green check-mark.

To add a document from a repository to an AWR platform project:

1. Select the document/data file you want to add and click the **Add to Project** button on the toolbar.

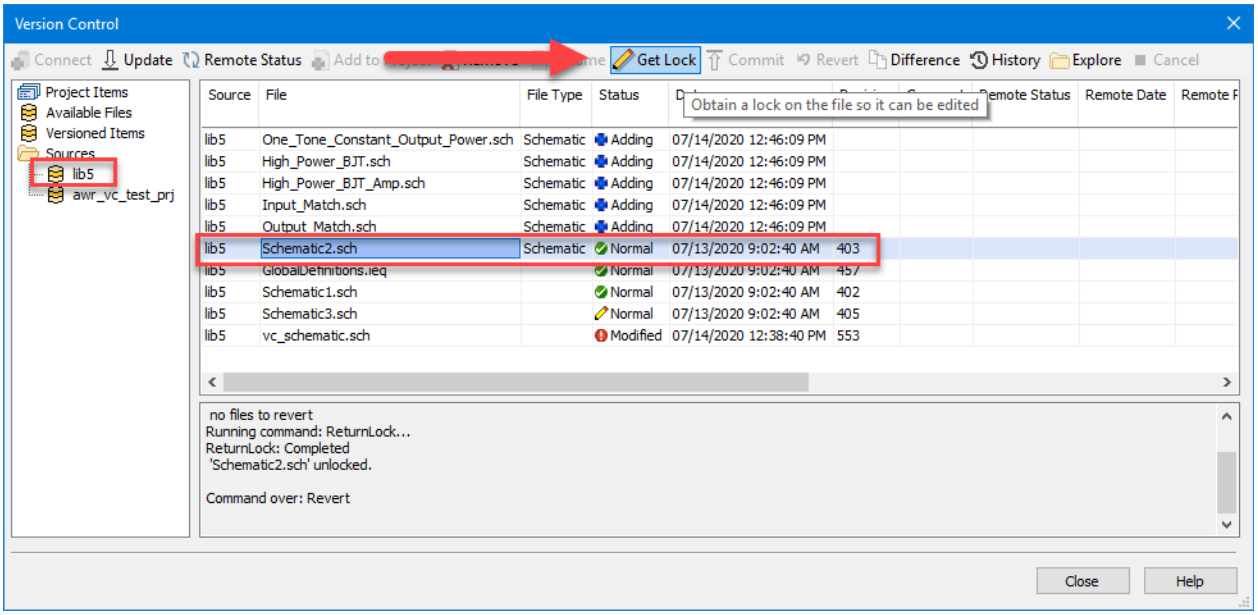


To add documents with PDK dependencies:

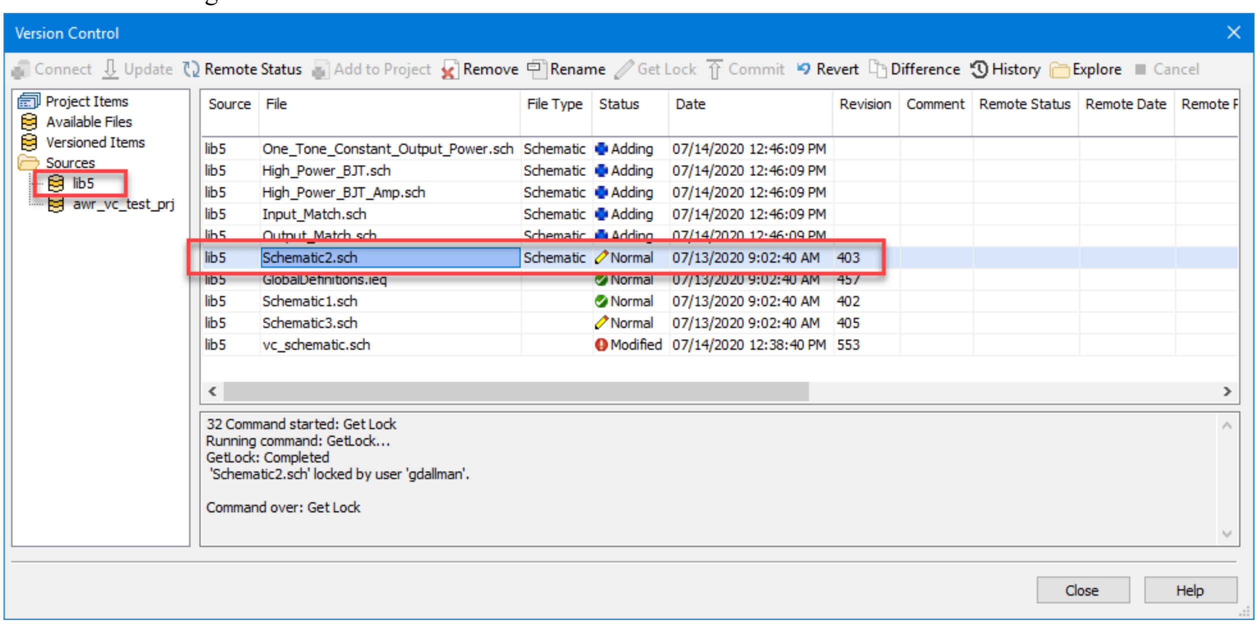
1. Ensure that the correct LPF and PDK versions are added in the project prior to adding a document from the repository to the AWR project.
2. Select the documents/data files you want to add and click the **Add to Project** button on the toolbar. Note that the Version Control dialog box does not manage PDK or LPF dependencies. You must ensure that all projects are using the correct LPF and PDK versions for their documents.

To edit a document under version control:

1. Select the document/data file you want to edit and click the **Get Lock** button on the toolbar.

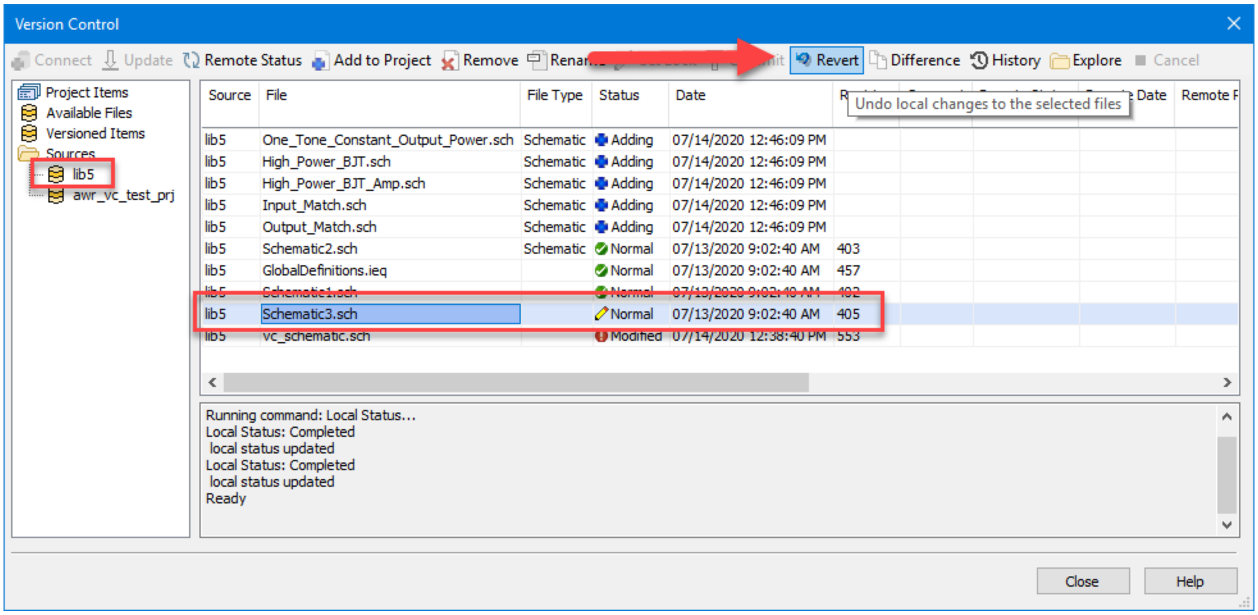


2. Add a minimum 5-character comment when prompted, then click **OK**.
3. The document/data file displays with a "Normal" **Status** and a pencil icon, indicating that the document is locked and available for editing.

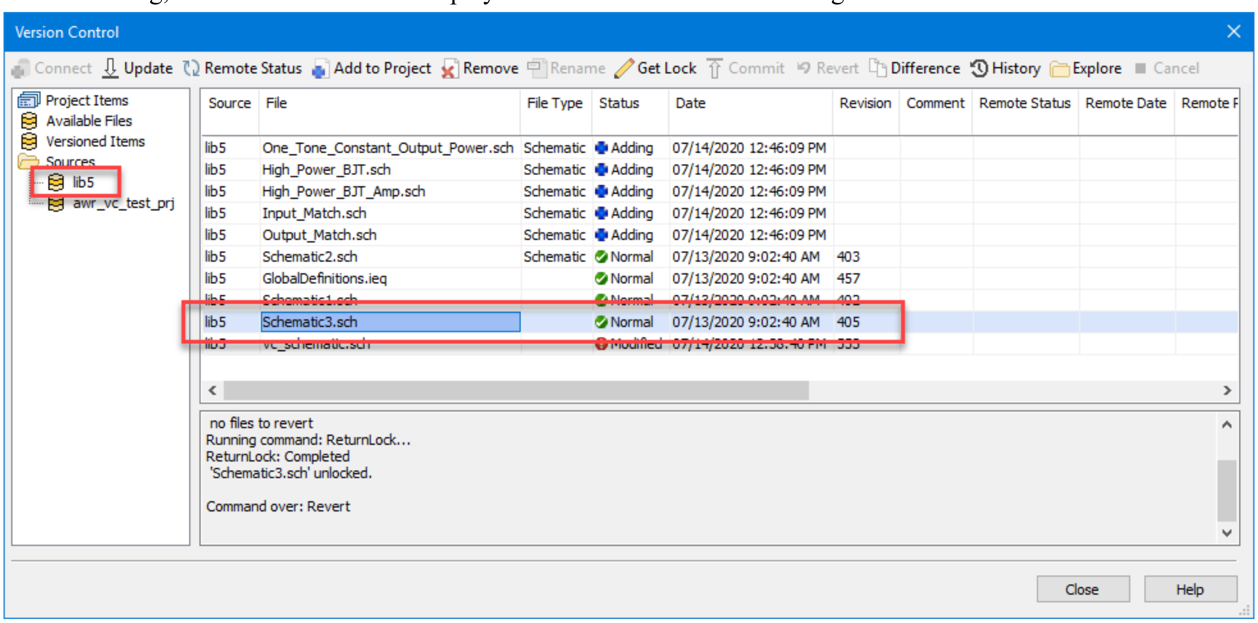


To revert a change to a document under version control:

1. Select the document/data file you want to revert and click the **Revert** button on the toolbar.

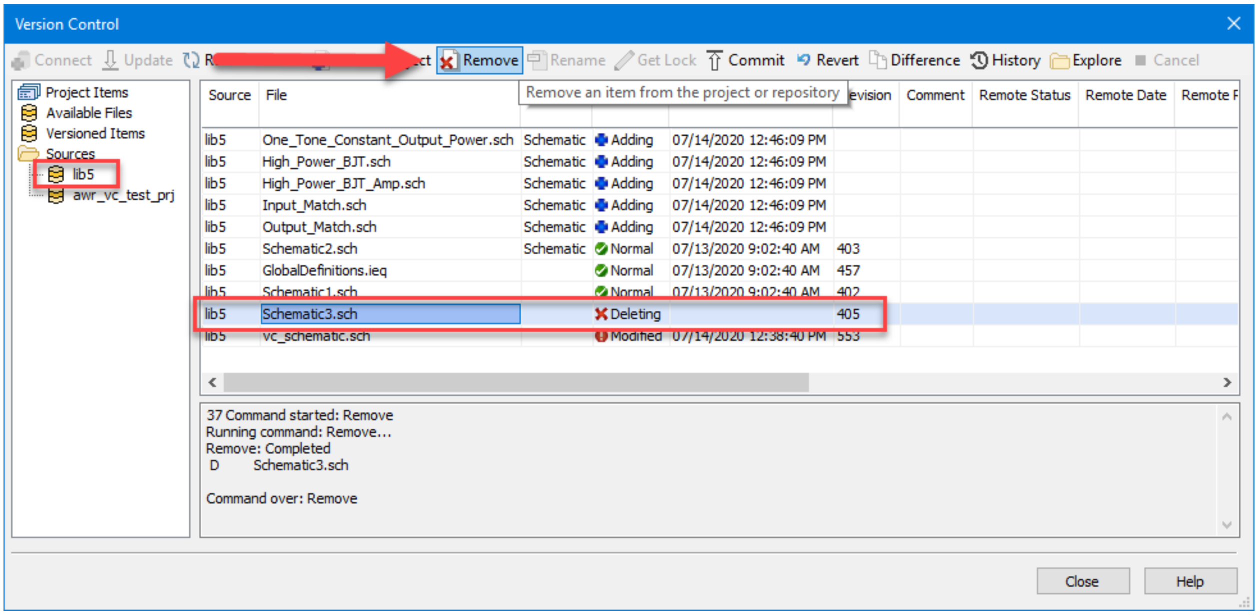


2. After reverting, the document/data file displays with a "Normal" **Status** and a green check-mark.

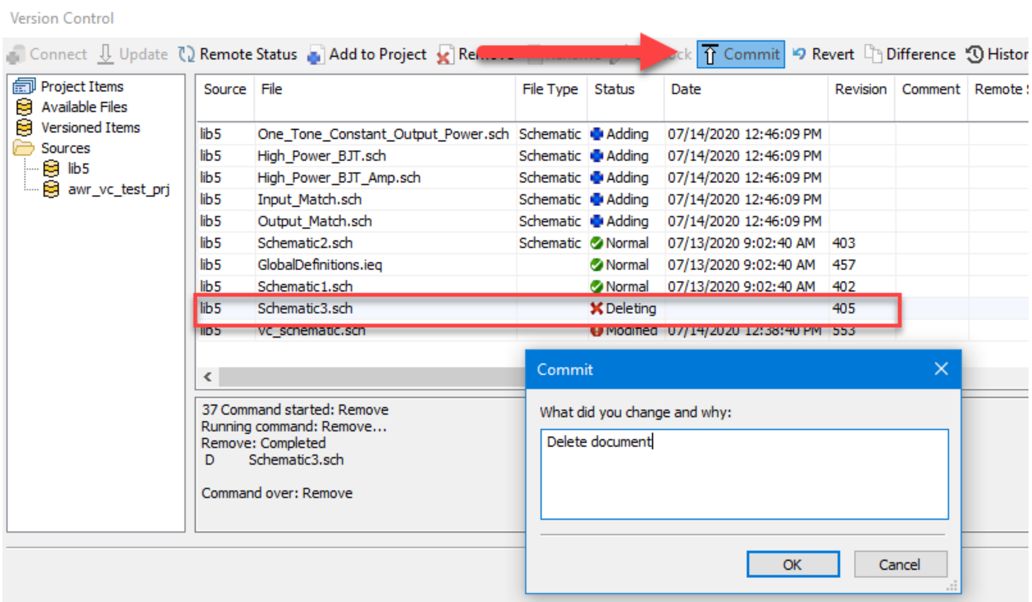


To remove a document from a repository:

1. Select the document/data file you want to remove and click the **Remove** button on the toolbar.



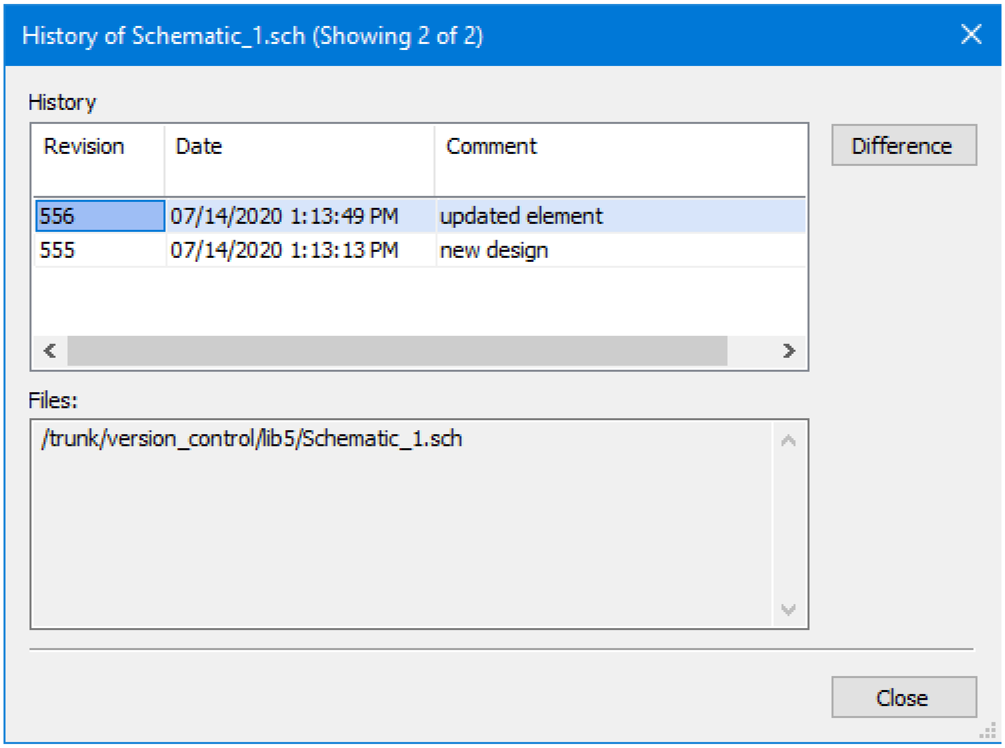
2. Select the document/data file again and it displays with a "Deleting" **Status** and a red "x".



3. Commit the document/data file to complete the deletion.

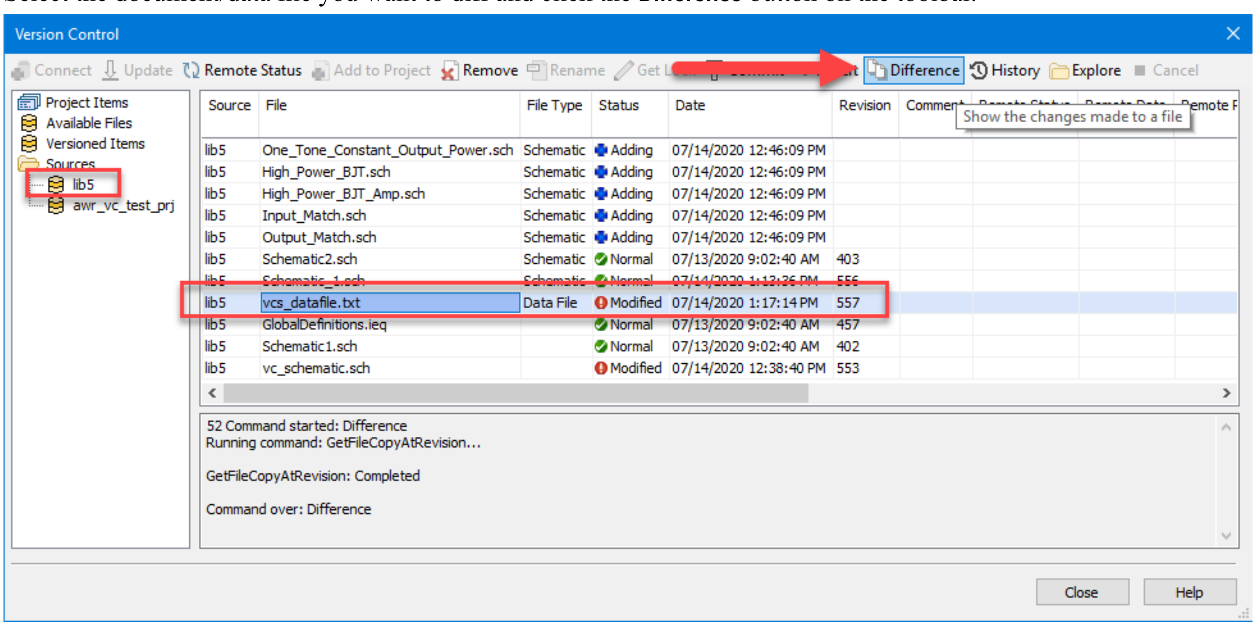
To check a document version history:

1. Select the document/data file you want to check and click the **History** button on the toolbar.
2. A History of <Document Name> dialog box displays with the document version history.



To perform a Difference operation on a document:

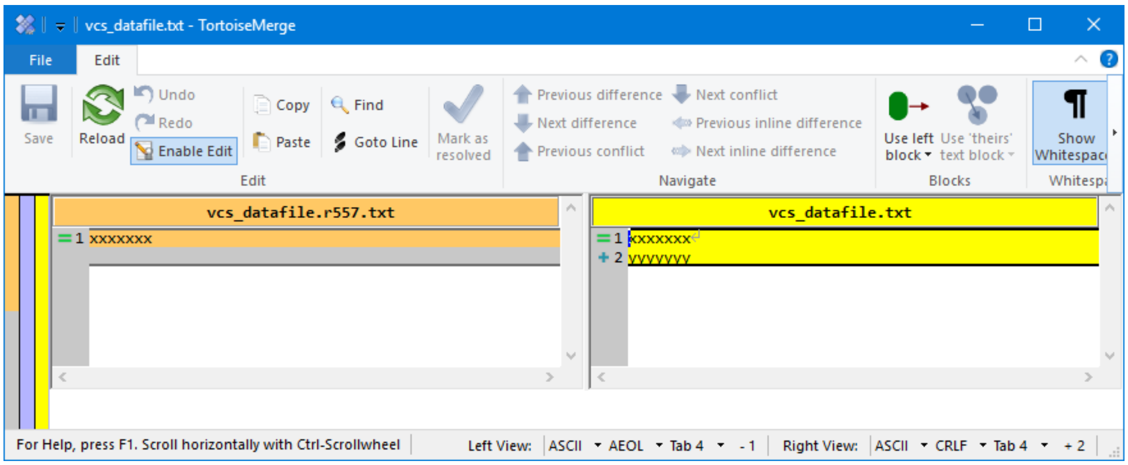
1. The **Difference** command in the Version Control dialog box is executed on a modified (but not committed) document to show the differences between the last committed version of the document and the modified (but not committed) document.
2. Select the document/data file you want to diff and click the **Difference** button on the toolbar.



To change the Difference tool from the default TortoiseSVN (*tortoise.exe*), in your *user.ini* file (choose **Help > Show Files and Directories** to locate it), add the following text:

```
[Versioncontrol Difference Tools]
default=fullpath: specify the full path of the tool to use for difference of supported
 vcs docs
filetype=fullpath or executable: specify the preferred difference tool for a file
extension, where filetype is JSON or other diff file
```

3. The previously checked in version of the document displays on the left of the comparison and the current working version displays on the right.



2.9.5. Version Control Troubleshooting

The following are the most common issues related to using version controlled documents/data files.

1. Unable to connect to the repository, or the repository is not populating.
   - **Subversion**: Check to see if you selected the command line option during Tortoise SVN installation. You can check the subversion installation directory to verify you have an *svn.exe* file.
   - **ClioSoft**:
     - Verify that you are connected to your company network.
     - Verify that you have the environment variable set.
     - Check your ClioSoft Server and Windows Client version numbers. The Windows ClioSoft Client version cannot exceed the ClioSoft Server version, or the SOS Client does not connect to the repository.
2. Where is the local repository folder located?
   - The repository is accessible by clicking the **Explore** button on the Version Control dialog box toolbar.
3. I imported a schematic and do not see my PDK information.
   - You must manage any PDK dependencies outside of the Version Control dialog box.

Chapter 3. Data Files

Data files are most commonly used for importing data (from simulation or measurement) for use during simulation. Data files can be either raw data format files, text data format files, Touchstone® format files, DC-IV format files, DSCR format files, or MDIF format data files. This chapter discusses how to work with data files and includes information specific to each data file type. An open data file window displays the data type in the window title.

3.1. Working With Data Files

You can add new data files to a project, or you can import existing data files for use as subcircuits, or simply link to a data file without importing it. All data files are editable; you can view, edit, delete, rename, and export the files.

3.1.1. Importing Data Files

You can import data files to use as subcircuits in schematics, for comparison purposes, or for any other purpose. (See "Importing Data Files Describing Subcircuits".)

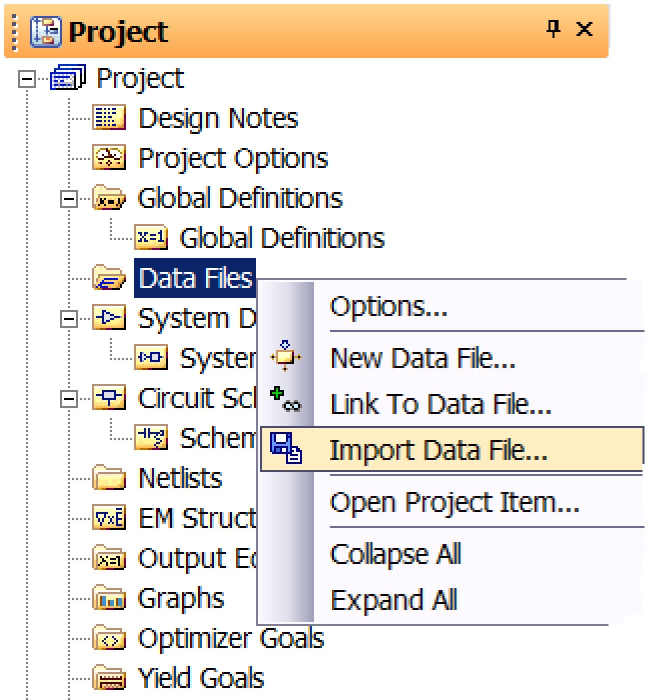
Imported data files are typically S-parameter files or another type of file that contains frequency domain N-port parameters. For more information on data file formats see "Data File Formats".

Each data file that you import is represented as a subnode under **Data Files** in the Project Browser, and exhibits the following naming scheme.

NOTE: The path to the data file displays only after you link to the file.

To import a data file into a project:

1. If you are adding a raw data file format, you must first set up the proper options for importing the file. Right-click the **Data Files** node in the Project Browser and choose **Options**. The Data File Options dialog box displays. Click the **Raw Data Format** tab to specify the format of the data, then click **OK**. For more information about this dialog box, see "Data File Options Dialog Box: Raw Data Format Tab".
2. Perform one of the following:
   - Choose **Project > Add Data File > Import Data File**.
   - Right-click **Data Files** in the Project Browser, and choose **Import Data File** to display the Open dialog box.



3. Select the desired data file, specify the format of the file in **Files of Type** , and then click **Open**. The data file displays as a subnode of **Data Files** in the Project Browser.

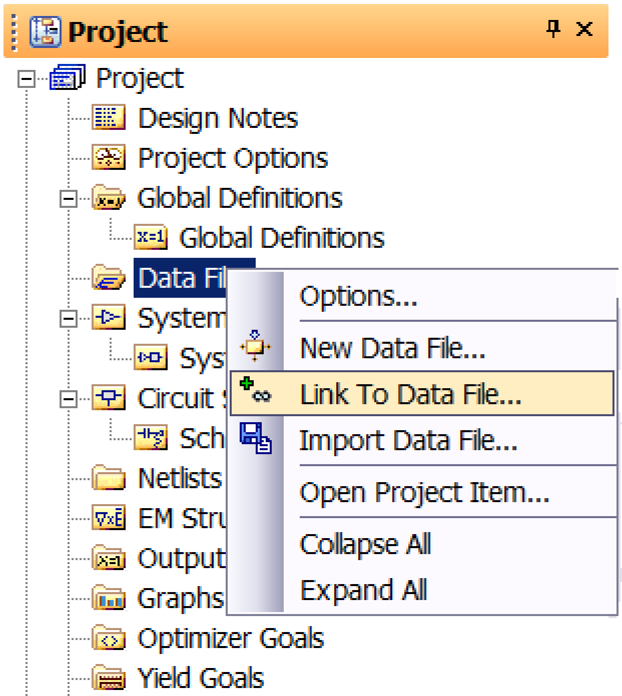
You can also import a data file through the Windows Explorer by displaying both the Windows Explorer and Cadence® AWR Design Environment® platform program windows. In the Windows Explorer, click on a data file (for example, a **.s2p* file) and drag and drop the file into the open AWR Design Environment platform program window. The file is automatically added to the project. You can import multiple files by pressing the **Ctrl** or **Shift** keys while performing this operation. When data files are added to the project using this method, the file extension determines the file format. If the file extension does not correctly indicate the format, you should add the data files by choosing **Project > Add Data File > Import Data File** as previously described.

A data file must have the proper file extension to be imported as a certain format. You may need to change a file extension to properly import the file. For the required file extension for each format type see the following file format sections.

3.1.2. Linking to Data Files

You can link to a data file instead of importing it. All of the details discussed in "Importing Data Files" apply except for a few items.

- Choose **Project > Add Data File > Link To Data File**.
- Right-click **Data Files** in the Project Browser, and choose **Link To Data File**.

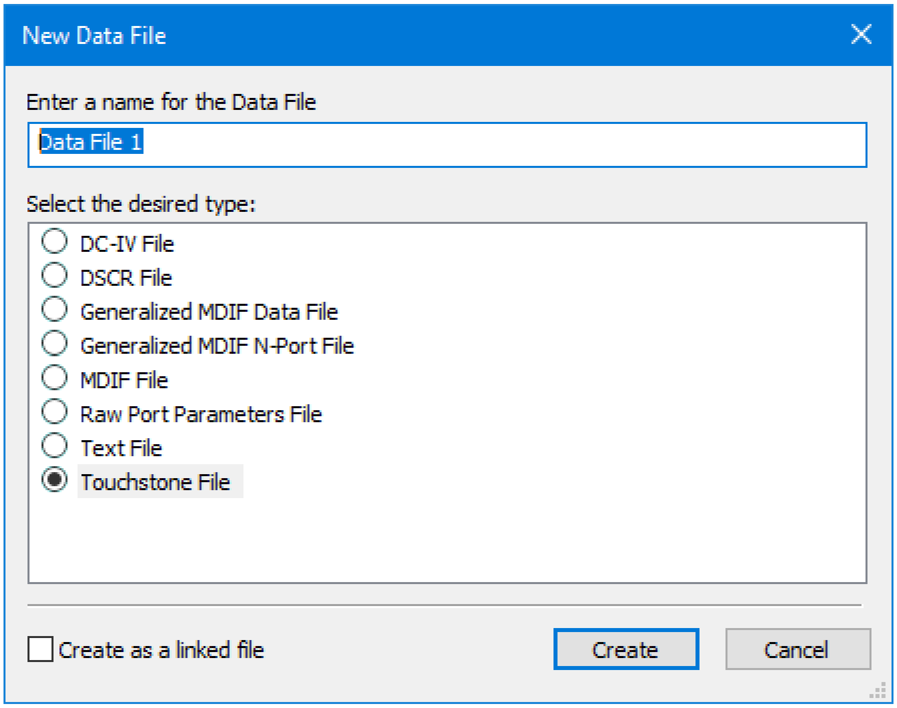


The data file is not saved with the project when linked.

You can right-click a linked data file and choose **Read Only** to toggle on and off read-only properties for the file. When a file is read-only, you can view the file but you cannot edit it.

3.1.3. Adding New Data Files

You can add a new, empty data file to a project by choosing **Project > Add Data File > New Data File**, or by right-clicking **Data Files** in the Project Browser and choosing **New Data File**. You specify the data file type in the New Data File dialog box.



You can copy and paste text into these new files or type in the text.

3.1.4. Editing Data Files

After you add, import, or link to data files you can perform the following operations:

Operation	Action
View a data file	Double-click the **Data Files** node in the Project Browser to open the Data File Options dialog box with options for viewing and editing the data file.
Edit a data file	You can edit the text in the Data File window using standard Windows editing commands such as Select, Cut, Copy, and Paste. The Data File Editor functions the same as the Windows Notepad. For imported data files, edits are implemented at the next simulation. For linked data files, you must save the data file before the change is effective after simulation. When you edit a linked data file and close the window, you are prompted to save the edits to the file on disk.
Delete a data file	Select the **Data Files** node in the Project Browser and choose **Edit > Delete** or press the **Delete** key. You can also right-click **Data Files** and choose **Delete Data File**. You are prompted to confirm deletion unless you press the **Shift** key while choosing **Delete Data File**.
Rename a data file	Right-click the **Data Files** node and choose **Rename Data File**.
Close a data file	Click the **X** button at the upper right of the window.
Export a data file	Right-click the **Data Files** node and choose **Export Data File**. In the Save As dialog box specify the name and location of the exported file.

3.2. Data File Formats

The AWR Design Environment platform supports the following data file formats. Information specific to each format is presented.

3.2.1. DC-IV Data File Format

The DC-IV data file format is a Cadence Microwave Office® program-specific format used for reading DC-IV curves. You can use this format to display measured or simulated IV curves. You can also use it in conjunction with simulated IV curves and the IVDELTA measurement to help optimize a model to match measured IV curves.

The following rules apply to DC-IV data files:

- Files must have a **.ivd* extension.
- The "**!**" character is used for comments, and comments may be inserted only before the data. Comments persist until the end of the line.
- The IV data must be complete; any empty items in the data matrix produce an error.
- The units for DC-IV files are always amps.

The following is the IVD data format.

```
m
n
   C1   ... Cn
X1 Y11 ... Y1n
. .   .
. .   .
. .   .
Xm Ym1     Ymn
```

where:

- `m` = The total number of swept values (typically the x-axis and IV plot)
- `n` = The total number of stepped values (typically the family of curve value)
- `C1 to Cn` = The values for the steps; these are the identifiers for the IV curve steps
- `X1 to Xm` = The x-values for the graph
- `Y11 to Ym1` = The y-values for the graph for the first stepped value
- `Y1n to Ymn` = the y-values for the graph for the nth stepped value

The following example shows the format of the data file.

```
8
4
0.0025   0.005   0.0075   0.01
1   8.5180e-02  9.2429e-02  9.4740e-02  9.5573e-02
2   1.5191e-01  2.1142e-01  2.1916e-01  2.2175e-01
3   1.5228e-01  2.6085e-01  3.3270e-01  3.4631e-01
4   1.5243e-01  2.6114e-01  3.5031e-01  4.2712e-01
5   1.5258e-01  2.6141e-01  3.5066e-01  4.2814e-01
```

```
6   1.5273e-01  2.6167e-01  3.5102e-01  4.2857e-01
7   1.5289e-01  2.6193e-01  3.5137e-01  4.2900e-01
8   1.5304e-01  2.6219e-01  3.5172e-01  4.2943e-01
```

3.2.2. DSCR Data File Format

In the DSCR data file format the data is in rows and columns. The values available for each parameter are arranged in columns. The BEGIN DSCR DATA line is followed by the % format line that specifies the names of dependent variables. The first column is always treated as a string; other columns are real, integer, or string, depending on the first row of data.

The first column, under the Index heading in the following example, contains entries used to identify each row in the file. These entries can be an integer or an alphanumeric identifier, and can be considered a list of specification numbers (or part numbers). For example, the data file data/stdvalues15.dscr is arranged as follows:

```
REM stdvalues15.dscr
BEGIN DSCRDATA
% INDEX A12 A13
1 1000 1000
2 1000 1200
3 1000 2200
4 1200 1000
5 1200 1200
6 1200 2200
END DSCRDATA
```

After the data file is imported, you can access the rows/columns of the data file the same way you access the text data files. You can plot directly to a graph using the PlotRow and PlotCol measurements, and using them in an equation with the DataFile and DataFileCol built-in functions.

3.2.3. Generalized MDIF Data File Format

The AWR Design Environment platform supports the Generalized Measurement Data Interchange Format (GMDIF) for arbitrary blocks of data, as well as load pull specific subsets. Arbitrary data blocks allow for graphical visualization of any data in the GMDIF formatted file, and can be used with measurements in the Data category. The following is a syntax example of an arbitrary data block and rules for the formatting.

The following rules apply to GMDIF data files:

- Files must have an **.mdf* or **.mdif* extension.
- The "!" character is used for comments, which you can insert anywhere in the data file. Comments persist until the end of the line.
- GMDIF files cannot have more than 7 parameters, or an error is issued.
- GMDIF supports any number of ports.
- The AWR Design Environment software can interpolate between GMDIF variables if they are defined as numeric type.
- The format line in GMDIF specifies the order of the data columns.

```
VAR var1 = 0                        ! First parameter value
```

```
VAR var2 = 0                          ! Second parameter value
Begin ARB1                            ! Arbitrary data block name (can be user defined)
%INDEX   DataName1 DataName2          ! Data format line, must provide Index and data
column names
1 0.5 0.75                            ! Data values
2 0.25 0.35
End

VAR var1 = 0
VAR var2 = 1
Begin ARB1
%INDEX   DataName1 DataName2
1 1.5 1.75
2 1.25 1.35
End
```

For a description of the load pull subset formats of GMDIF, see “Load Pull Specific GMDIF Formats”

3.2.4. Load Pull Specific GMDIF Formats

To use this load pull functionality, the load pull data must be written as a generalized MDIF data file that conforms to relatively strict conventions (this restriction enables a much simpler use model for end users). Until load pull vendors can write this format natively, conversion scripts are provided to convert the various load pull files into this format. Two types of load pull files are supported. The first type specifies the measured a and b waves taken from the measurements. This format is preferable because it provides the most flexibility in use. The second type of file writes final measured quantities directly (such as output power, or PAE).

3.2.4.1. A/B Wave Format

The following are design decisions to consider for A/B wave format:

- Any number of swept dimensions can be supported, but there is a fixed set of 'inner' sweeps that must conform to a standard convention to allow the software to intelligently manipulate the data. These sweeps are:
  - Swept Impedances, over source or load and at one or more harmonics
  - Sweeping over more than one variation adds a dimension to the data set produced
  - Swept input power
  - Swept fundamental frequency
- The VAR variables that are repeated for each MDIF data block should only represent variables that are swept in an outer sweep.
  - Any fixed quantities will be written in a header block

Impedance Sweeps

Since the actual measured quantities of interest are computed from the A\B waves, the sweeps over values that are computed from the A\B waves are represented by integer indexes. Sweeps over impedance values are represented by a single index dimension into all the impedance values (so 1d sweep for the 2d impedance data). The variable names used to represent the swept quantities must conform to guidelines for correct interpretation. The following shows all of the valid swept variable names that you can use to represent an impedance sweep in the MDIF file. To make the indexes consistent with the UI, the range should start from 1.

- iGammaL1 (index into gamma load values at the fundamental frequency)

  - iGammaL1 is required as a listed variable in the header even if the quantity is not swept
- iGammaL2 (index into gamma load values at the 2nd harmonic frequency)
- iGammaL3 (index into gamma load values at the 3rd harmonic frequency)
- iGammaS1 (index into gamma source values at the fundamental frequency)
- iGammaS2 (index into gamma source values at the 2nd harmonic frequency)
- iGammaS3 (index into gamma source values at the 3rd harmonic frequency)

Power Sweeps

Since the input available power is computed from the A/B waves, you use an index to represent the power sweep (to avoid the ambiguity of potentially mismatched data, and to also keep the data regular if input power varies over the impedance points).

- iPower (index into the swept input powers)
  - iPower is required as a listed variable in the header even if the quantity is not swept

Frequency Sweeps

The fundamental frequency can be swept. If it is swept, a real valued sweep is used for the frequency. The value should always be written in Hz.

- F1 (fundamental frequency - if the fundamental frequency is not swept, then this value should be included in the HEADER block)

Arbitrary Sweeps.

Any variable can be swept but there are some naming requirements. If the quantity being swept can be derived from the A/B waves (e.g. power) or DC voltages/currents then the variable name must be prefixed with an “i”. If the quantity being swept cannot be derived from the A/B waves then the variable name must be prefixed with an “r”.

MDIF Data Blocks

The following data blocks must be used within the data file.

- HEADER - Use to write properties that are global to the entire file. Certain values in this block are required. The data is written as columns of values, with no VAR values for the block.
  - index - This is a dummy independent variable value, but you can use it to store multiple values in the header versus an index. Typically, there is only one row of values in the HEADER though, with an index value of zero.
  - F1(1) - Fundamental frequency, this should only be included if the data is not swept over the fundamental frequency (if it is swept, the values are in swept VAR value for the ABWAVES block)
  - GammaS1(3) - Source gamma at the fundamental harmonic. If not specified, 0.0 (matched to Z0SOURCE) is assumed. The values in the file are in two columns (real and imaginary). If a source pull is done the data is in the ABWAVES block, so this value should be omitted for that case.
  - GammaS2(3) - Source gamma at the 2nd harmonic. If not specified, 0.0 is used.
  - GammaS3(3) - Source gamma at the 3rd harmonic. If not specified, 0.0 is used.
  - Z0(3),Z0SOURCE(3),Z0LOAD(3) - Defines the characteristic impedance for the source and load. If Z0 is defined, then it is used for both Z0SOURCE and Z0LOAD. Any impedance that is not defined is assumed to be 50 ohms.
  - VERSION(2) - Optional string

  - DATE(2) - Optional string
  - TC(1) - Optional temperature in Celsius
- ABWAVES (represents the A/B waves swept over harmonics, if there are any. The DC data and source impedance data may also be included in this block). The column names for the data must be one of the following. Note that all complex values have a "(3)" as part of the name, and are represented as two columns of data each.
  - harm(1) - real value used to indicate the harmonic frequencies (real needed for multi-tone mixing products). The harmonic frequency is harm*F1
  - Vn(1) - DC voltage, where n is an integer, so V1, V2, and so on...
  - In(1) - DC current, where n is an integer, so I1, I2, and so on...
  - an(3) - complex a wave, where n is an integer, so a1 or a2
  - bn(3) - complex b wave, where n is an integer, so b1 or b2
  - GammaS(3) - complex source gamma for swept input impedance, this column is optional

You can use the following optional blocks within the data file:

- Data Information - This optional block is a departure from the standard MDIF format and is used to remove redundant information and reduce file size.
  - Contains independent variable names.
  - Contains ABWAVES block data column names and order.
  - Contains indicators of which data locations have data and which do not ("V" for "valid data" and "M" for "missing data").
  - Wrapping of data on rows is not supported, so a single row of data must be on a single line.

The following is a compact file format example of swept source impedance, swept power, and swept fundamental at 3 harmonics. It shows 80 gamma points and 10 power sweep points at 2 fundamental frequencies (most data omitted).

```
!Header block contains any globals
 BEGIN HEADER
 % index(0) NHARM(0) Z0(3)
 1 3 50 0
 END

 !Data information section provides the order of the independent
 !variables, the data column names, and indicators of which data locations
 !have data and which do not.
 !The "V" and "M" markers in the data locations indicate whether the data for that
 !row / column exists in the file.
 !For example, in this format the DC data is missing from any non fundamental
 !harmonic row
 VAR<> F1(1)
 VAR<> iGammaS1(0)
 VAR<> iPower(0)
 BEGIN<> ABWAVES
 % harm(1) a1(3) b1(3) a2(3) b2(3) V1(1) I1(1) V2(1) I2(1) GammaS(3)
 V V V V V V V V V V
 V V V V V M M M M V
 V V V V V M M M M V
 END<>
```

```
!Each ABWAVES block is at a single power, single load gamma, and value
!of any other independent variable
2200000000
1
1
1 0.016071 -0.044884 0.016069 -0.04488 -2.5955E-9 -9.2932E-10 -0.00054637
  0.001526 -0.3 -2.9985E-7 3 0.00031172 -0.7882 0.11756
2 -5.1706E-16 5.6353E-16 7.0943E-10 6.5099E-10 -8.1252E-11 9.8919E-11
  0.00012418 0.000102 0 0
3 4.5067E-17 6.5676E-17 1.3167E-10 -9.0948E-11 -2.0453E-17 -3.4216E-17
  -6.9098E-11 4.1305E-11 0 0

2400000000
1
1
1 0.016071 -0.044884 0.016069 -0.04488 -2.3655E-9 -8.4697E-10 -0.00054637
  0.001526   -0.3 -2.9985E-7 3 0.00031172 -0.7882 0.11756
2 -4.6804E-16 5.0985E-16 7.0943E-10 6.5099E-10 -7.3566E-11 8.9561E-11
  0.00012418 0.000102 0 0
3 4.0258E-17 5.8263E-17 1.3167E-10 -9.0948E-11 -2.633E-17 -4.3811E-17
  -9.8466E-11 5.9177E-11 0 0

:
:
2400000000
80
10
1 0.0068691 -0.030403 0.0068685 -0.0304 -1.7581E-9 -3.9723E-10 -0.00023354
  0.0010336 -0.3 -2.9992E-7 3 0.00029871 -0.8882 0.19021
2 -2.7557E-16 5.4537E-16 6.8674E-10 3.4703E-10 -2.3525E-11 4.9402E-11
  6.2016E-5 2.9531E-5 0 0
3 1.4528E-16 1.2253E-16 2.4974E-10 -2.967E-10 -2.026E-17 -2.2086E-17 -4.4603E-11
  4.0916E-11 0 0
```

3.2.4.2. Derived Quantity Format

The format used for this type of load pull data attempts to conform to the A/B wave format. The following are design decisions to consider for derived quantity format:

- Since the derived quantities never make sense measured at the harmonics, the freq dimension is not part of this format.
- The input power can be the independent variable for each data block.
- To avoid the issue with the input power changing over different impedance values, a similar iPower index is used for the power sweep, and the actual input power is represented as dependent data value.
- It is important to use the same independent sweep names (for example, iGammaL1, iPower, or F1), but not all the dependent column values need to conform to a standard (the standardized derived names and values listed in the table that follows should be used for the recognized quantities).
- You should set the value for harm to allow for recovery of the harmonic/intermod frequency as described for the A/B wave file format.

Standard Derived Values

When possible, you should use the standard derived value names and unit conventions (as shown in the following table). Conforming to these conventions allows these values to be recognized and displayed with correct units, and also makes it possible to use the values in other automated calculations. It is important that any recognized quantity in the table is written in base units. If the values are not written as base units, the automatic assignment of unit types on read-in causes the values to be scaled incorrectly.

This optional block is a departure from the standard MDIF format and is used to remove redundant information and reduce file size. It contains independent variable names, LPDATA block data column names and order, and indicators of which data locations have data and which do not ("V" for "valid data" and "M" for "missing data").

The following is a compact file format example of swept load impedance, swept power, and swept fundamental at 3 harmonics. It shows 80 gamma points and 10 power sweep points at 2 fundamental frequencies (most data omitted).

```
!Header block contains any globals
BEGIN HEADER
% index(0) NHARM(0) GammaS1(3)
1 3 50.0 0.0
END

!Data information section provides the order of the independent variables,
!the data column names, and indicators of which data locations have data
!and which do not.
!The "V" and "M" markers in the data locations indicate whether the data
!for that row / column exists in the file.
!For example, in this format the DC data is missing from any non fundamental
!harmonic row
VAR<> F1(1)
VAR<> iGammaL1(0)
VAR<> iPower(0)
BEGIN<> LPDATA
% harm(1)  GammaL(3) PSrc_Ava(1)  PLoad(1)  G_Power(1)  PAE(1)
V          V         V            V         V           V
V          V         M            V         V           V
V          V         M            V         V           V
END<>

!Each LPDATA block is at a single power, single load gamma, and value of any
!other independent variable
1.5e9
1
1
1.  .555     .555   -10.0   12.676037   0.132988    12.203945
2.  .655     .755           13.626342   0.240446    13.067126
3.  .755     .755           14.254252   0.316895    14.516464

1.5e9
2
1
1.  .155     .155   -9.0    15.676037   0.432988    15.203945
2.  .255     .255           16.626342   0.540446    16.067126
3.  .355     .355           17.254252   0.616895    17.516464


:
:
2e9
80
10
1.  .555  .555   5.0     12.676037  0.832988 12.203945
```

```
2.  .555  .555   0       12.626342  0.840446 13.067126
3.  .555  .555   0       12.254252  0.816895 16.516464
```

Calculated Values

This table shows the calculated values that are automatically computed from the A/B wave format. For the derived value format, the values in the file should match the conventions (names and units) of the values in the table. The De-embed Support column shows which values can be de-embedded from a derived value load pull file. All de-embed operations require GammaL and PLoad. Additional values needed for other quantities are shown in the table.

Name	SPL Mapped Name	LPC Mapped Name	Derived Value File Support	Type	Unit	De-embed Support	Description
GammaL<harm>	gamma_ld<harm>	Gamma/Phase[deg]		complex	none	Yes	Reflection coefficient of the load
GammaS<harm>	gamma_src<harm>			complex	none	No (not applicable)	Reflection coefficient at the input of the device
PLoad	Pout_dBm	Pout[dBm]		real	dBW	Yes	Power delivered to the load
PLoadT	N/A	N/A		real	dBW	Yes	Total power delivered to the load, including harmonics
PDC				real	Watts	No	DC power dissipated in the device
PAE	Eff_%	PAEff[%]		real	%	Yes (requires PDC or PSrc_Del)	Value for 100% would be 100
Drain_Eff	Drain_eff	OutEff[%]		real	%	Yes (requires PDC or PSrc_Del)	Value for 100% would be 100
G_Trans	Gt_dB	Gain[dB]		real	dB	Yes	Transducer power gain
G_Power	Gp_dB			real	dB	Yes	Operating power gain
G_Compress			Yes	real	dB	No (not applicable)	Gain compression measured as the ratio of G_Trans over swept power

Name	SPL Mapped Name	LPC Mapped Name	Derived Value File Support	Type	Unit	De-embed Support	Description
							relative to the initial G_Trans value (linear, or lowest power gain).
G_CompressMG			Yes	real	dB	No (not applicable)	Gain compression measured as the ratio of G_Trans over swept power relative to the highest G_Trans value (max gain).
PSrc_Ava	Pin_avail_dBm	Psource[dBm]		real	dBW	No (not applicable)	Power available from the source
PSrc_Del	Pin_deliv_dBm	Pin[dBm]		real	dBW	No (not applicable)	Power delivered to the device from the source.
AMPM				real	Radians	No	Angle of b2/a1 (typically plotted over swept power)
AMPM_Offset				real	Radians	No	Angle of b2/a1 - Angle of b2/a1 at the lowest power sweep point
Compress_1db				real	dBW	Yes	Pload value at the 1db compression point (this measurement gives the same value at all power sweep points)
Compress_2db			Yes	real	dBW	Yes	Pload value at the 2db compression point (this measurement gives the same

Name	SPL Mapped Name	LPC Mapped Name	Derived Value File Support	Type	Unit	De-embed Support	Description
							value at all power sweep points)
IMD	C_up_dBm, C_lo_dBm, I3_up_dBm, I3_lo_dBm		Yes	real	dB	No	Intermodulation distortion measured as the ratio of Pload at the nearest fundamental to Pload of the selected harm value.[1]
IPN			Yes	real	dBw	No	Output intercept point measured as a function of Pload at the nearest fundamental to Pload of the selected harm value.[2]
IIPN			Yes	real	dBw	No	Input intercept point measured as a function of Pload at the nearest fundamental to Pload of the selected harm value.[3]

[1] IMD: Choosing the low side fundamental for the harm value gives the worst case third-order IMD of all possible tone combinations, and choosing the high-side fundamental for the harm value gives the best case third-order IMD of all possible tone combinations.

[2] IPN: The tone order (for example, 3rd order or 5th order) used in the output intercept point (IPN) calculation is automatically determined based on the selected harm value. Choosing the low-side fundamental for the harm value gives the worst case third-order IPN of all possible tone combinations, and choosing the high-side fundamental for the harm value gives the best case third-order IPN of all possible tone combinations. Note that the selected iPower value determines which power level is used for the intercept point calculation. Since intercept point is an extrapolation based on an amplifier operating in a linear range (where the fundamental output power increases 1 dB with a 1 dB increase in input power and the third-order intermod product output power increases 3 dB with a 1 dB increase in input power) selecting an iPower value higher than 1 is risky. This functionality is included so that IPN can be calculated correctly with measured data that is dynamic range limited on the intermod products (and, thus, for lower power levels the assumptions about a 1 dB increase in fundamental input power do not lead to a 3 dB increase in third-order intermod power).

[3] IIPN: Input intercept Point (IIPN) is calculated as IPN - Transducer Gain of the fundamental. The tone order (for example, third-order or fifth-order) used in the intercept point calculation is automatically determined based on the selected harm value. Choosing the low-side fundamental for the harm value gives the worst case third-order IIPN of all possible tone combinations, and choosing the high-side fundamental for the harm value gives the best case third-order IIPN of all possible tone combinations. Note that the selected iPower value determines which power level is used for the intercept point calculation. Since intercept point is an extrapolation based on an amplifier operating in a linear range (where the fundamental output power increases 1 dB with a 1 dB increase in input power and the third-order intermod product output power increases 3 dB with a 1 dB increase in input power) selecting an iPower value higher than 1 is risky. This functionality is included so that IIPN can be calculated correctly with measured data that is dynamic range limited on the intermod products (and, thus, for lower power levels the assumptions about a 1 dB increase in fundamental input power do not lead to a 3 dB increase in third-order intermod power).

3.2.5. Generalized MDIF N-Port File Format

The AWR Design Environment platform supports a second subset of the MDIF (Measurement Data Interchange Format) file format call the Generalized MDIF format (GMDIF). [1] The AWR Design Environment platform GMDIF files, which have a *.mdf* extension, allow importing S-parameters which vary with frequency and with one or more named parameters. They are used in conjunction with parameterized subcircuits, in which the subcircuit's parameter names and values are automatically assigned to match those contained within the GMDIF file when the subcircuit is associated with the file. You can create these text files manually using any text editor, or with automated tools capable of producing GMDIF files as output.

The following rules apply to GMDIF data files:

- Files must have a **.mdf* extension.
- The "!" character is used for comments, which you can insert anywhere in the data file. Comments persist until the end of the line.
- GMDIF files cannot have more than 7 parameters or an error is issued.

MDIF and GMDIF formats differ as follows:

- GMDIF supports any number of ports. MDIF only supports 2 ports.
- The AWR Design Environment software can interpolate between GMDIF variables if they are defined as numeric type. MDIF format cannot be interpolated.
- The format line in GMDIF specifies the order of the data columns. (In MDIF the order is fixed, independent of the format line.)

For GMDIF files, the VAR definitions have different types-- integer, double, and string. For the VAR settings in the files, you can specify types several ways:

- (0) is integer
- (1) is double
- (2) is string
- If not set and no quotes, then number
- If not set and quotes, then string

The following are examples of the different types. The first two are double type and the last two are string type.

[1] User-modifiable default. Modify by editing under $DEFAULT_VALUES in the *default.lpf* file in the root installation directory.

- VAR Vc_mA(1) = 30
- VAR Vc_mA = 30
- VAR VC(2) = 30mA
- VAR VC = "30mA"

The type only matters if you want to interpolate between the values. To interpolate, the values must be double or integer type.

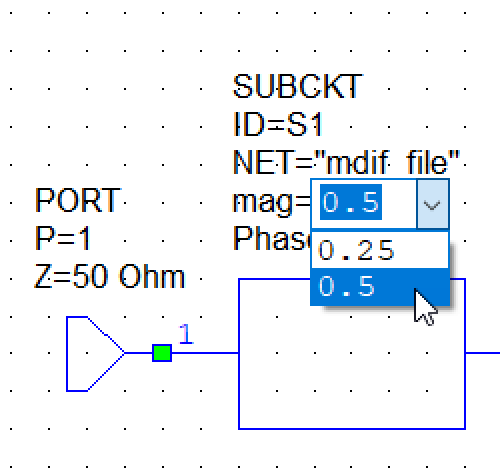
3.2.5.1. Using GMDIF in a Schematic

When using a GMDIF file in a schematic, by default, the values are all the discrete values in the file. If a combination of values does not have data, a simulation error results. The following simple MDIF file is an example.

```
VAR mag=0.25
VAR Phase=0
Begin ACDATA
# GHz S MA R 50
% F   N11X  N11Y
1 0.25 0
End

VAR mag=0.25
VAR Phase=180
Begin ACDATA
# GHz S MA R 50
% F   N11X  N11Y
1 0.25 180
End

VAR mag=0.5
VAR Phase=0
Begin ACDATA
# GHz S MA R 50
% F   N11X  N11Y
1 0.5 0
End
```

When used in a schematic, the values available for the "mag" parameter are 0.25 and 0.5 and the "Phase" parameter is 0 and 180. Notice that there is not a block defined for mag=0.5 and Phase=180. If you set the parameters to these values, a simulation error results.

By default, you select the discrete values set in the MDIF file as shown in the following figure.



You can enable interpolation such that you can enter any numeric value or assign to a variable and the results are interpolated between the actual data points. You can change the interpolation settings globally by choosing **Options > Project Options** to display the Project Options dialog box, clicking the **Interpolation/Passivity** tab, and then selecting **Enable parameter interpolation**. You can make the same setting local to each data file by right-clicking the data file and choosing **Options** to display the Project Options dialog box, clicking the **Interpolation/Passivity** tab, and then ensuring that the **Use project defaults** check box is cleared to allow you to select **Enable parameter interpolation**.

3.2.6. MDIF File Format

The AWR Design Environment platform supports a subset of the MDIF (Measurement Data Interchange Format) file format. The AWR Design Environment platform MDIF files, which have a *.mdf* extension, allow importing S-parameter and noise figure data which varies with frequency and with one or more named parameters. They are used in conjunction with parameterized subcircuits, in which the subcircuit's parameter names and values are automatically assigned to match those contained within the MDIF file when the subcircuit is associated with the file. You can create these text files manually using any text editor, or with automated tools capable of producing MDIF files as output.

The following rules apply to MDIF data files:

- Files must have a **.mdf* extension.
- The "**!**" character is used for comments, and comments may be inserted anywhere in the data file. Comments persist until the end of the line.
- MDIF files only support two-port files.

When MDIF files are used in schematics, the blocks of data are sorted by various rules. You can change the order by right-clicking the file in the Project Browser under the **Data Files** node, choosing **Options**, and then clicking the **MDIF Files** tab to change the sorting rules. For more information on this dialog box, see "Data File Options Dialog Box: MDIF Files Tab ".

3.2.6.1. MDIF File Structure and Syntax

An MDIF file consists of one or more data blocks. Each data block is associated with one or more named parameters (independent variables). Data blocks can refer to S-parameter data or optional noise figure data. Basic MDIF syntax contains four reserved words:

- VAR begins an independent variable definition line, in the form VAR<name>=<value>. You can use VAR statements to specify data that varies with one independent variable, or to specify multidimensional data, (data that varies with two or more independent variables). A VAR statement must have a name on the left side; the value on the right side can be a number or an alphanumeric string.

- BEGIN <blockname> signals the beginning of a data block.
- END signals the conclusion of a data block.
- REM or the "**!**" character at the beginning of a line signifies a comment.

You can create the MDIF file by manually combining separate S-parameter (*.s2p*) files, provided the required MDIF format elements are inserted appropriately. Multiple sets of data (ACDATA and NDATA) are used with VAR statements before each data block. The file data set is made up of one or more such data blocks, each separated by a BEGIN and END statement. For *.mdf* files, only ACDATA and NDATA are allowed.

The MDIF file format follows:

```
REM MDIF Basic Syntax Example
    VAR TEMP = value1   /! first parameter, first value
    VAR AGC = value_a   /! second parameter, first value
    BEGIN ACDATA     /! required AC data block
    ....
    ....ACDATA block data lines
    ....
    END
    BEGIN NDATA        /!optional noise data block
    ....             /!same parameter values as above
    ....ACDATA block data lines
    ....
    END
    VAR TEMP = value2     /!first parameter, second value
    VAR AGC = value_b     /! second parameter, second value
    BEGIN ACDATA
    ....
    ....  block data lines ....
    ....
    END
```

The following is the ACDATA block and NDATA block for various data points (2-port with 50-ohm S-parameters):

```
 BEGIN ACDATA
 # AC ( GHZ  S  DB  R  50  FC 1  0  )
  %F  n11x   n11y  n21x  n21y  n12x  n12y n22x  n22y
 !  RF-freq S11-db S11-deg S21-db S21-deg S12-db S12-deg S22-db S22-deg
  1.0000  -15 45 -8 25 -20 -12 -15 10
  2.0000  -16 25 -9 30 -20 -12 -15 20
  3.0000  -17 -10 -10 35 -20 -12 -11 30
 END
 BEGIN NDATA
 # GHz           S          MA           R         50
  %F  nfmin n11x n11y rn
  1  2.50  0.7  -5   190
  2  2.60  0.75  -8  180
  3  2.70  0.8  -12  170
 END
```

The option line syntax "`# AC ( GHZ S DB R 50 FC 1 0 )`" sets frequency units to GHz, 2-port parameters to S, 2-port parameter format to dB, reference impedance to 50 ohms, and output frequency equal to the input frequency

(Fout = 1* Fin + 0). You can also use option line syntax identical to *.SnP* files. This is demonstrated in the following example. The format line "`% F n11x n11y n21x n21y n12x n12y n22x n22y`" specifies how the column ordering of the file is associated with the elements of the S-parameter matrix. Ordering must be identical to the *.S2P* file format regardless of the contents of the format line. Reference impedance in the option line must be 50 ohms. For 10-port and more the column names should display as "`% F n11x n11y n21x n21y n12x n12y n22x n22y ..........n9_10x n9_10y n9_11x n9_11y.....n10_1x n10_1y n10_2x n10_2y...`"

3.2.6.2. Complete MDIF File Example

The following shows an example of a complete MDIF file:

```
!Single parameter MDIF Datafile
!Shows .S2P-style option line syntax
VAR Vg = -1
BEGIN ACDATA
# GHz           S          DB          R         50
 %F  n11x  n11y  n21x  n21y  n12x  n12y  n22x  n22y
 10  -0.091484   60.432  24.417   -117.09   -77.086  66.176   -2.9307    73.43
 15  -0.10407   -91.037  24.933     63.704  -76.552 -111.41   -3.749    -155.06
 20  -0.096309   49.786  23.801   -118.37   -77.66   68.134   -3.9473    88.734
END
BEGIN NDATA
# GHz           S          MA           R         50
 %F  nfmin n11x n11y rn
 10  1.2  0.6  50   30
 15  1.3  0.65  45  40
 20  1.4  0.67  40   50
END
VAR Vg = 0
BEGIN ACDATA
# GHz           S          DB          R         50
 %F  n11x  n11y  n21x  n21y  n12x  n12y  n22x  n22y
 10    -0.096681   60.419  27.417   -119.16   -78.057  64.121  -5.3158   76.657
 15    -0.11014   -91.038  27.684     66.848   -77.772 -108.24  -6.7222  -162.43
 20    -0.10081    49.771  26.501   -121.75   -78.93   64.792  -7.0243   97.234
END
BEGIN NDATA
# GHz           S          MA           R         50
 %F  nfmin n11x n11y rn
 10  2.2  0.8  70   110
 15  2.3  0.81  71  122
 20  2.4  0.82  72   135
END
VAR Vg = 1
BEGIN ACDATA
# GHz           S          DB          R         50
 %F  n11x  n11y  n21x  n21y  n12x  n12y  n22x  n22y
 10     -0.63853  -146.02  -1.0349   -134.35   -76.251  45.718   -9.0561  86.985
 15     -0.52831   26.267  -1.7256     40.229  -76.941  -139.66  -10.646   174.8
 20     -0.97244  -130.81  0.68099   -125.8     -74.535  54.346 -10.764  123.07
END
BEGIN NDATA
# GHz           S          MA           R         50
 %F  nfmin n11x n11y rn
 10  1.8  0.5  -180   220
```

```
 15  1.9  0.55  -171  133
 20  1.7  0.64  -166   43
END
```

3.2.7. Raw Data File Format

The raw data format is used to read N-port network data files written as rows and columns of data in a text file. The raw data format provides an easy method for importing data from spreadsheets, math programs, or test equipment. It is also useful for importing files that are close to, but not true Touchstone format files.

The format of the data in the rows and columns is specified by right-clicking the **Data Files** node in the Project Browser and choosing **Options** to display the Data File Options dialog box. Click the **Raw Data Format** tab to specify the format of the data, then click **OK** . For more information about this dialog box, see "Data File Options Dialog Box: Raw Data Format Tab".

The following rules apply to raw data files:

- Files must have a **.prn* extension.
- A "**!**" character is used for comments, and comments are only allowed before the data if you are specifying one matrix per line. Otherwise, you cannot include any comments in the file.
- The numbers in a row of data must be separated by spaces or tabs.
- All raw data files in a single project must use the same raw data file format.
- The raw data format does not support noise parameters.

The following example demonstrates a sample 2-port data file read using one matrix per line, row major data, and a real imaginary format:

```
f1 ReS11 ImS11  ReS12 ImS12  ReS21 ImS21  ReS22 ImS22
f2 ReS11 ImS11  ReS12 ImS12  ReS21 ImS21  ReS22 ImS22
f3 ReS11 ImS11  ReS12 ImS12  ReS21 ImS21  ReS22 ImS22
```

The same example using column major data order displays as:

```
f1 ReS11 ImS11  ReS21 ImS21  ReS12 ImS12  ReS22 ImS22
f2 ReS11 ImS11  ReS21 ImS21  ReS12 ImS12  ReS22 ImS22
f3 ReS11 ImS11  ReS21 ImS21  ReS12 ImS12  ReS22 ImS22
```

If the size of the matrix is specified (instead of using the one matrix per row option) then the first example could be written as:

```
f1    ReS11 ImS11 ReS12 ImS12 ReS21 ImS21
ReS22 ImS22 f2   ReS11 ImS11 ReS12 ImS12
ReS21 ImS21 ReS22 ImS22 f3 ReS11 ImS11
ReS12 ImS12 ReS21 ImS21 ReS22 ImS22
```

Because of the limitations of this file format, Cadence recommends using the AWR Design Environment platform to convert to a true Touchstone file format using the following procedure:

1. Import the raw data file.
2. Plot some or all of the data to make sure the data displays correctly on a graph. If it doesn't, you may need to adjust your raw data settings.

3. Once the data displays correctly, add an output file to the project by right-clicking the **Output Files** node in the Project Browser, choosing **Add Output File**, and then selecting **Port Parameter** with the new raw data file specified as the data source name.
4. Simulate, and the AWR Design Environment software outputs a Touchstone formatted file on your computer that you can import as a Touchstone file.

3.2.8. Text Data File Format

The text data file format is used by various system simulator models and supported by the `vfile()` equation function. You can plot data from text files on graphs using the PlotCol and PlotRow measurements. You can use this method to compare simulated versus measured data, like a power sweep, for example. Finally, you can use data from text files for model parameters using the Data and Row or Col functions. A typical application is to set your PORT_ARBS bit sequence in a text file.

The data file is an ASCII text file comprised of three sections that must display in the following order:

1. Tags (optional)
2. Column Headings (optional)
3. Column Data

3.2.8.1. Comments

Sections of the file may be 'commented out'. Comments are ignored when the data file is interpreted.

There are two types of comments: line comments and block comments. Line comments ignore the remaining text on the line on which they appear. Block comments ignore text until an end marker is detected.

Line comments begin with a "**!**" character:

```
 ! This is a line comment
SMPFRQ=10 G     Sampling frequency of 10 GHz
```

All text from the "**!**" character to the end of the line are ignored.

Block comments begin with "**/***" characters and end with "***/**" characters:

```
/* This starts the comment block.
SMPRATE=16
The above line is ignored. This ends the comment block. */
```

Comment characters that are in quotations are not treated as comments, for example:

```
TITLE="Don't use this file"
```

3.2.8.2. Tags

The first section of the file, if present, is the Tags section. Tags are 'name' 'value' pairs that provide additional information about the file. Each tag consists of a name followed by an "=" followed by a value. Numeric values can also be followed by an optional units scale.

```
SMPFRQ = 10.0 G   Indicates sampling frequency of 10 GHz
```

```
SMPRATE = 8

TITLE = "Sample Data"
```

The name consists of an alphabetic character followed by one or more alphabetic characters, digits, or the "_" character. The alphabetic characters can be either upper or lower case; case is not considered when interpreting tags.

Values are either numeric, in which case they use standard numeric form such as 1.0, 1e9 or 5.2e-10, or text. Text values must be enclosed in quotation marks "". A quotation mark may be included as part of the text by preceding it with a "\" character.

```
TITLE = "From \"The Big Book\""
```

Numeric values can be followed by one of the following units scales:

Unit Abbreviation	Unit name	Description
f	femto	10^{-15}
p	pico	10^{-12}
n	nano	10^{-9}
u	micro	10^{-6}
m	milli	10^{-3}
c	centi	10^{-2}
k	kilo	10^{3}
M	mega	10^{6}
G	giga	10^{9}
T	tera	10^{12}
mil	0.001 inches	
in	inch	
ft	feet	
mile	miles	
C	Celsius	
K	Kelvin	
F	Fahrenheit	
rad	radians	
deg	degrees	
dbm	dBm	
dbw	dBW	

Numeric values may also be entered in hexadecimal (base 16) format by preceding the value with 0x. For example, 0x12 represents the decimal value 18.

The following is the set of predefined tags:

Tag Name	Tag Description	Tag Type
SMPFRQ	Sampling frequency	Numeric

Tag Name	Tag Description	Tag Type
TSTEP	Time step	Numeric
SMPRATE	Sample rate	Numeric
SMPSYM	Samples per symbol	Numeric
CTRFRQ	Center frequency	Numeric
MEASFRQ	Measurement frequency	Numeric
Z0	Impedance	Numeric
T0	Start time	Numeric
NROWS	Number of rows	Numeric
NCOLS	Number of columns	Numeric

You can specify only one SMPFRQ or TSTEP, since they are related by TSTEP = 1/SMPFRQ.

If NROWS is specified, no more than NROWS of data are read from the file. This is useful when testing data sets.

If NCOLS is specified, no column headings should be specified. NCOLS is normally used to indicate how many columns are represented by interleaved data that appears in a single column or row.

You can also specify tags not in the predefined list. These tags are available to the individual models. Tags not used by a model are ignored.

3.2.8.3. Column Headings

The column headings section, if present, provides additional information about the data columns of the file. If this section is present, you must specify a column heading for each data column. A column heading has the following format:

```
[name]([type][,units])
```

where `[name]` is the name of the column, `[type]` is the type of data and `[units]` is the units for the data. Each of the parts is optional, although the "(,)" and "," punctuation are required.

Each column heading is separated from the others by one or more spaces or a tab character.

The column name has the same restrictions as a tag; it must start with an alphabetic character followed by zero or more alphabetic characters, digits or underscores "_". It cannot contain spaces.

The column type indicates how complex values are to be generated, and can be one of the following:

Column Name	Column Type
Re	Real component
Im	Imaginary component
I	Inphase component
Q	Quadrature component
Mag	Magnitude component
Phs	Phase component
Scalar	Non-complex data.

The column types are not case-sensitive.

If the column type is not Scalar, the column must be followed by a corresponding matching column containing the other component of the complex value. The following are the pairs:

- Re, Im or Im, Re
- I, Q or Q, I
- Mag, Phs or Phs, Mag

The units for the data are SI units and several common units such as the following:

Unit Abbreviation	Unit Name
GHz	gigahertz
ns	nanoseconds
mW	milliwatts
dBm	DBm
dBW	dB Watts
Deg	angle in degrees
Rad	angle in radians

The following is an example of a frequency response file, with a column of frequency values in GHz followed by a column of magnitude values in dBm, followed by a column of phase values in degrees:

```
Freq(,GHz) (Mag,dBm)  (Phs,deg)
1          10         20
2          11         23
```

Note that in this example the frequency column does not use a column type, while the magnitude and phase columns do not use column names.

The following illustrates how you can use the Scalar column type:

```
(Scalar) (Mag,dBm)    (Phs,deg)
1        10           20
2        11           23
```

The following is illegal, since an Re column must be followed by an Im column. Use the Scalar column type as in the previous example instead to specify complex values with only a real component and 0 for the imaginary:

```
(Re)     (Mag,dBm)     (Phs,deg)
1        10            20
2        11            23
```

3.2.8.4. Column Data

The last section is the column data. The column data consists of numeric data values separated by spaces, tabs, or commas. If neither an NCOLS tag nor column headings are used in the data file, the number of columns is determined by the number of values on the first line of data.

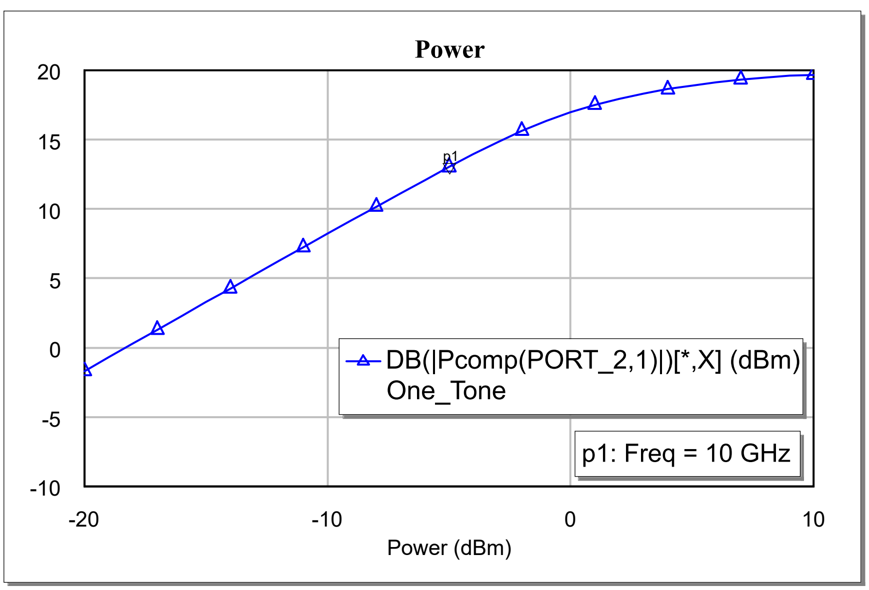
If an NCOLS tag or column headings are specified, the columns are defined by the NCOLS tag or by the column headings, and the data values do not have to appear in the specified columns. For example, if you have three sets of data values interleaved into a single column, you could simply add the following to treat the data as three columns of data:

```
NCOLS=3
1
11
.01
2
12
.02
3
13
.03

This is the equivalent of:

NCOLS=3
1    11    .01
2    12    .02
3    13    .03
```

If an NROWS tag is specified, only the equivalent of that many rows of data is processed. This is useful when you have a large set of data, but only want to use a small portion of it for testing purposes.

3.2.8.5. Use with Microwave Office

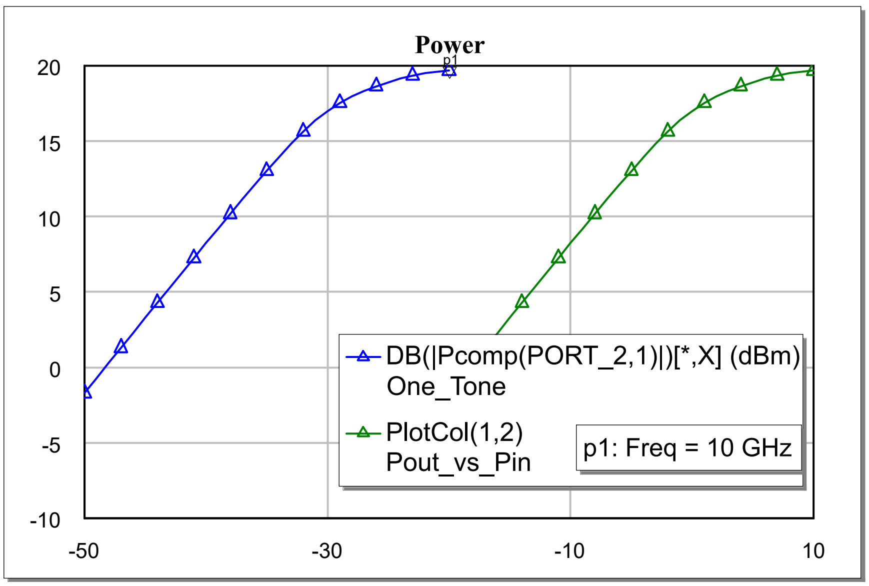
You can use text data files to plot data from measurements or other computer programs versus simulation results. A common problem is not using column headers correctly to line up the data. This problem is demonstrated in the following example of a plot of output power versus input power. The following graph shows the simulation result.



The following data file shows the original plotted data for the first few points:

```
! AM to PM characteristics
-20 -1.70501
-19 -0.705099
-18 0.293446
-17 1.29193
-16 2.2832
-15 3.27812
-14 4.27169
-13 5.26358
-12 6.25331
-11 7.2403
-10 8.22375
```

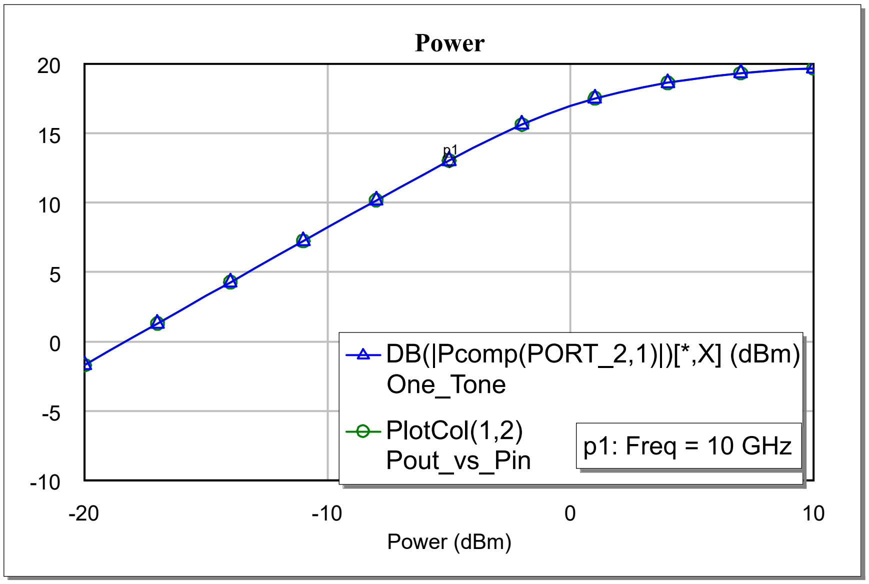
When this data is plotted versus the simulated data using the PLOTCOL measurement, the data displays as follows.



As you can see, the data is shifted by 30 dB. The solution is to add the proper column headers. After changing the data file to the following,

```
! AM to PM characteristics
Pin(,dBm) Pout(,dBm)
-20 -1.70501
-19 -0.705099
-18 0.293446
-17 1.29193
-16 2.2832
-15 3.27812
-14 4.27169
-13 5.26358
-12 6.25331
-11 7.2403
-10 8.22375
```

the data lines up with the simulated data, as shown in the following graph.



There are several items to note when plotting data in this mode:

- The text before the "( )" is not used. You can add text for identification purposes or omit it.
- Complex data is not used in this mode. You can omit the type (no text after the first "(" and before the ","). For example, Pin(,dBm).
- The unit modifier for the y-axis does not currently change how the data is plotted. However, once you use headers, you must add them for each column, so Cadence recommends that you define the units for each column of data.

The following examples show other common data types:

Current in mA

```
Pin(,dBm) ID(,mA)
-20 33.0683
-19 33.0857
-18 33.1075
-17 33.135
-16 33.1696
-15 33.2132
-14 33.2679
-13 33.3369
-12 33.4236
-11 33.5327
-10 33.6697
```

Frequency in GHz versus impedance in ohms

```
(,GHz) (,ohms)
```

```
1 166.824
2 93.9818
3 72.9005
4 63.8995
5 59.2724
6 56.6005
7 54.9267
8 53.8125
9 53.0351
10 52.4719
```

Voltage in volts and time in seconds

```
Voltage(,V) Time(,s)
1.9870509463574 0
-1.8883353991916 250.40064102564e-12
1.7804170729503 500.80128205128e-12
-1.6639524028713 751.20192307692e-12
1.5396674521328 1.0016025641026e-9
-1.408352913535 1.2520032051282e-9
1.2708585477709 1.5024038461538e-9
-1.1280871015799 1.7528044871795e-9
0.98098775371646 2.0032051282051e-9
```

3.2.9. Text Data File Load Pull and Source Pull Formats

The AWR Design Environment platform can also import load pull and source pull data as text files. Maury Microwave and Focus Microwaves file formats are supported, as is the format exported by the Load Pull script (see “Load Pull Script”).

Imported load pull files are used in the program in two ways. First, you can plot the data from the load pull files using the measured Load Pull measurements. When adding a Load Pull measurement you can plot:

- *LPCM*: Measured load pull contours
- *LPGPM*: Measured load pull impedance points
- *LPCMMax*: Maximum of measured load pull contours
- *LPCMMin*: Minimum of measured load pull contours
- *LPINT*: This measurement determines the impedance of a linear network and then finds the right point within the load pull data to return the Load Pull measurement data. For example, if based on the impedance of your matching network, it can show the expected output power or PAE based on the load pull data.

You can also use the load pull points in the files as the impedances used during simulated load pull so your simulated load pull uses the same impedances as your measured load pull.

3.2.9.1. Maury File Formats

The AWR Design Environment platform supports version3, version4, and version5 of the Maury load pull file format.

3.2.9.2. Swept Power Files

Load pull systems also produce files where the tuner impedances are fixed and the input power is swept. You can perform the following:

1. Import the file as a text file.
2. Comment out any lines in the file, except the column headers.
3. Put the column headers in the right units format. See [“ Column Headings ”](#). This step is important or the data might not line up properly versus simulation. For example, for power, the typical unit is dBm. If you do not change the column header to specify the proper units, the program uses base units (dBw in this example) and there is a 30 dB shift in the data.

3.2.10. Touchstone File Format

The Touchstone file format allows data to be read in as G-, H-, S-, Y-, or Z-parameters.

Touchstone-compatible data files are comprised of a header that describes the format of the network parameter matrices.

The header syntax is:

```
>#   HZ|KHZ|MHZ|GHZ|THZ  G|H|S|Y|Z  MA|DB|RI   [R x]
```

Where the "`|`" character separates different choices, and the "`[ ]`" brackets indicate an optional entry. The following table lists each item in the header:

<table>
<tr><th>Header Portion</th><th>Description</th></tr>
<tr><td><code>#</code></td><td>Signifies the beginning of the header.</td></tr>
<tr><td><code>HZ | KHZ | MHZ | GHZ | THZ</code></td><td>Specifies the frequency units of the data file (choose one).</td></tr>
<tr><td><code>G | H | S | Y | Z</code></td><td>Specifies the parameter type of the data file (choose one).</td></tr>
<tr><td><code>MA | DB | RI</code></td><td>Specifies how the complex data are presented (choose one).</td></tr>
<tr><td><code>[R x]</code></td><td>x is a real number that specifies the reference impedance (optional).</td></tr>
</table>

The following are example headers:

```
# GHZ S MA R 50
# MHZ S DB
# HZ Z RI
```

The following is network data syntax, where `m` specifies the number of frequency points, and `n` specifies the matrix size:

```
<freq point 1> <row 1>
[<row 1 cont.>]
<row 2>
[<row 2 cont.>]
......
<row n>
<row n cont.>]
<freq point 2> <row 1>
[<row 1 cont.>]
<row 2>
[<row 2 cont.>]
......
<row n>
[<row n cont.>]
...
<freq point m> <row 1>
```

```
[<row 1 cont.>]
<row 2>
[<row 2 cont.>]
......
<row n>
[<row n cont.>]
```

Noise data can be added to two-port data files and must follow port parameter data. The first frequency point in the noise data must be less than or equal to the highest frequency point in the port parameter data. The following is the noise data format:

```
Freq NFmindB MagOpt AngOpt Rn
```

where

- `Freq` is the frequency of noise data in frequency units.
- `NFmindB` is the minimum noise figure in dB.
- `MagOpt` is the magnitude of the normalized source gamma to achieve `NFmin`.
- `AngOpt` is the angle (in degrees) of the normalized source gamma to achieve `NFmin`.
- `Rn` is the normalized noise resistance. To have a physical meaning, `Rn` must be greater than or equal to

$$\frac{NFmin - 1}{4 \cdot Re(Yopt)}$$

  where `NFmin` is the minimum noise factor (not in dB) and `Yopt` is the optimum source admittance. If `Rn` is less than this minimum it is reset to the minimum.
- `GammaOpt` and `Rn` are normalized to the reference impedance specified in the header for the port parameter data (usually 50 ohms).

The following rules apply to Touchstone format files:

- Files must have a **.g??,*.h??,*.s??,*.y??,*.z??* extension. The file name extensions, by convention, are s1p, s2p, ..., s9p, s10p, s11p - s99p. The same convention is used for y- and z- file extensions. These extensions correspond to 1- through 99-port data files; however, the extension is not used to determine the size of the network parameter matrices. Instead, the first network parameter matrix is read from the file and its size is computed and used for the remaining network parameter matrices, so the maximum size of a readable network parameter matrix is only limited by your hardware.
- The "**!**" character is used for comments, which may be inserted anywhere in the data file. Comments persist until the end of the line.
- The reference impedance is used as the normalizing impedance for all network parameters. For S-parameters, it is Z0. For Y-parameters, the y-matrix is divided by R. For Z-parameters, the z-matrix is multiplied by R. For G-parameters g(1,1) is divided by R and g(2,2) is multiplied by R. For H-parameters, h(1,1) is multiplied by R and h(2,2) is divided by R. If no reference impedance is specified, then 50 ohms is assumed.
- G- and H-parameters are supported for two-port files only.
- T-parameters are not supported for import, but you can plot T-parameters from networks.
- MA and DB indicate that the complex data is in polar form (magnitude, angle), the angle of which is always in units of degrees; DB further specifies that the magnitude has been transformed via 20*Log(mag). RI indicates that the data is in rectangular form (real, imag).

- The network parameter matrices are in row major order, except for two-port matrices, which are in column major order.
- Each network parameter is a complex number that is read as two sequential real numbers.
- Each line may contain a maximum of four network parameters (8 real numbers). If the matrix contains more than four network parameters per row (it is larger than a four-port), the remaining network parameters are continued on the following line.
- Each row of the network parameter matrices begins on a new line.
- The first row of a network parameter matrix is preceded by the frequency at which the data was generated.

The following is an example file for a one-port:

```
 # GHZ S RI R 50
f1 ReS11 ImS11
f2 ReS11 ImS11
f3 ReS11 ImS11
```

The following is an example file for a two-port:

```
 # GHZ S RI R 50
f1 ReS11 ImS11  ReS21 ImS21  ReS12 ImS12  ReS22 ImS22
f2 ReS11 ImS11  ReS21 ImS21  ReS12 ImS12  ReS22 ImS22
f3 ReS11 ImS11  ReS21 ImS21  ReS12 ImS12  ReS22 ImS22
```

Note that two-port files use column major format and allow more than one row of the matrix on one line.

The following is an example file for a three-port:

```
 # GHZ S RI R 50
f1 ReS11 ImS11  ReS12 ImS12  ReS13 ImS13
   ReS21 ImS21  ReS22 ImS22  ReS23 ImS23
   ReS31 ImS31  ReS32 ImS32  ReS33 ImS33
f2 ReS11 ImS11  ReS12 ImS12  ReS13 ImS13
   ReS21 ImS21  ReS22 ImS22  ReS23 ImS23
   ReS31 ImS31  ReS32 ImS32  ReS33 ImS33
f3 ReS11 ImS11  ReS12 ImS12  ReS13 ImS13
   ReS21 ImS21  ReS22 ImS22  ReS23 ImS23
   ReS31 ImS31  ReS32 ImS32  ReS33 ImS33
```

The following is an example file for a four-port:

```
 # GHZ S RI R 50
f1 ReS11 ImS11  ReS12 ImS12  ReS13 ImS13 ReS14 ImS14
   ReS21 ImS21  ReS22 ImS22  ReS23 ImS23 ReS24 ImS24
   ReS31 ImS31  ReS32 ImS32  ReS33 ImS33 ReS34 ImS34
   ReS41 ImS41  ReS42 ImS42  ReS43 ImS43 ReS44 ImS44
f2 ReS11 ImS11  ReS12 ImS12  ReS13 ImS13 ReS14 ImS14
   ReS21 ImS21  ReS22 ImS22  ReS23 ImS23 ReS24 ImS24
   ReS31 ImS31  ReS32 ImS32  ReS33 ImS33 ReS34 ImS34
   ReS41 ImS41  ReS42 ImS42  ReS43 ImS43 ReS44 ImS44
f3 ReS11 ImS11  ReS12 ImS12  ReS13 ImS13 ReS14 ImS14
   ReS21 ImS21  ReS22 ImS22  ReS23 ImS23 ReS24 ImS24
```

```
   ReS31 ImS31  ReS32 ImS32  ReS33 ImS33 ReS34 ImS34
   ReS41 ImS41  ReS42 ImS42  ReS43 ImS43 ReS44 ImS44
```

The following is an example file for a five-port:

```
 # GHZ S RI R 50
f1 ReS11 ImS11  ReS12 ImS12  ReS13 ImS13 ReS14 ImS14
   ReS15 ImS15
   ReS21 ImS21  ReS22 ImS22  ReS23 ImS23 ReS24 ImS24
   ReS25 ImS25
   ReS31 ImS31  ReS32 ImS32  ReS33 ImS33 ReS34 ImS34
   ReS35 ImS35
   ReS41 ImS41  ReS42 ImS42  ReS43 ImS43 ReS44 ImS44
   ReS45 ImS45
   ReS51 ImS51  ReS52 ImS52  ReS53 ImS53 ReS54 ImS54
   ReS55 ImS55
f2 ReS11 ImS11  ReS12 ImS12  ReS13 ImS13 ReS14 ImS14
   ReS15 ImS15
   ReS21 ImS21  ReS22 ImS22  ReS23 ImS23 ReS24 ImS24
   ReS25 ImS25
   ReS31 ImS31  ReS32 ImS32  ReS33 ImS33 ReS34 ImS34
   ReS35 ImS35
   ReS41 ImS41  ReS42 ImS42  ReS43 ImS43 ReS44 ImS44
   ReS45 ImS45
   ReS51 ImS51  ReS52 ImS52  ReS53 ImS53 ReS54 ImS54
   ReS55 ImS55
f3 ReS11 ImS11  ReS12 ImS12  ReS13 ImS13 ReS14 ImS14
   ReS15 ImS15
   ReS21 ImS21  ReS22 ImS22  ReS23 ImS23 ReS24 ImS24
   ReS25 ImS25
   ReS31 ImS31  ReS32 ImS32  ReS33 ImS33 ReS34 ImS34
   ReS35 ImS35
   ReS41 ImS41  ReS42 ImS42  ReS43 ImS43 ReS44 ImS44
   ReS45 ImS45
   ReS51 ImS51  ReS52 ImS52  ReS53 ImS53 ReS54 ImS54
   ReS55 ImS55
```

See Touchstone® File Format Specification for Touchstone v2.0 file format information.

3.2.10.1. Specifying Port Names in Touchstone Data Files

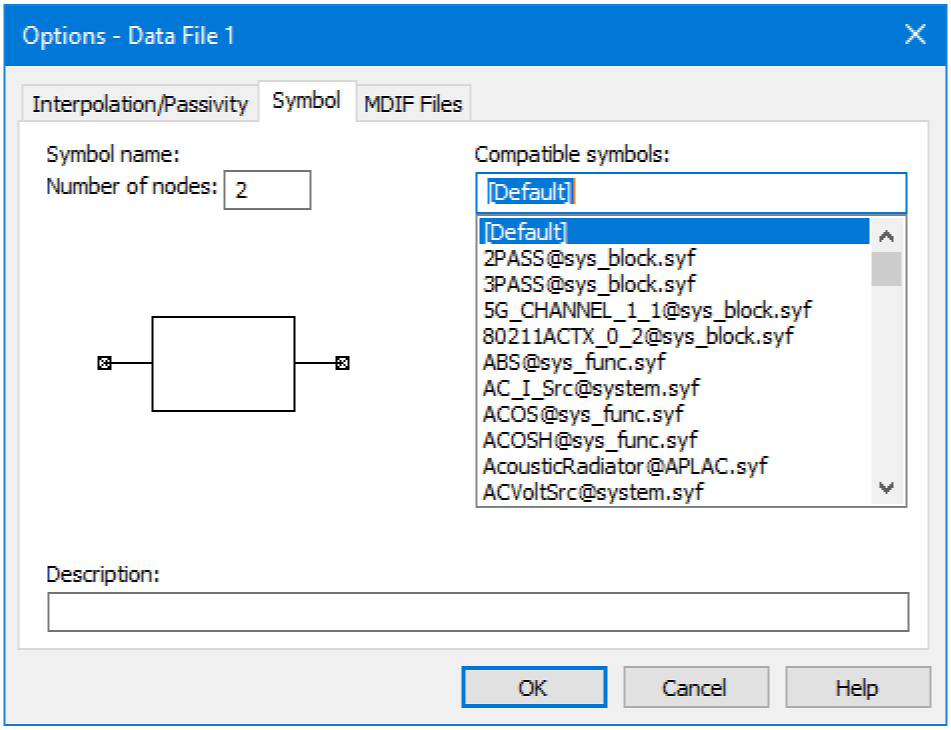
You can specify port names in comment lines (which begin with an exclamation point character) in the data files:

```
     ! Port[1]=In
     ! Port[2]=Out

     # HZ S MA R 50
     ! Freq MagS11 AngS11 MagS21 AngS21 MagS12 AngS12 MagS22 AngS22
     0 0.047460043 -0 0.777978 0 0.77797801 -0 0.1831028 180
     1e+009 0.076247202 48.161048 0.77658837 -3.4250583 0.77658838 -3.4250583 0.18476243
 168.16514
```

The port index is specified inside the brackets, and the name is provided after the equal sign, without quotes.

3.2.10.2. Port Names On SUBCKT Schematic Symbols

Port names automatically display next to the terminals of the SUBCKT element unless disabled for a specific Touchstone file on the **Symbol** tab of the Options dialog box.



3.2.10.3. NPORT_F Output File Measurement

When you use an NPORT_F Output File measurement to create a Touchstone data file, if the referenced data source is a schematic that contains ports with PIN_IDs, those names are written into the data file using the syntax previously described.

3.3. Advanced Data File Topics

The following sections include information about converting to and working with Touchstone format data, extrapolation, and using data files with noise simulation.

3.3.1. Citi Format Files

Some network analyzers use Citi format to store their measurement data. The AWR Design Environment platform does not directly import Citi data, but has scripts that can convert Citi files to Touchstone format, including multiple-parameter Citi files. See the Cadence website for example Citi import scripts, or contact Cadence AWR Support to request these scripts.

3.3.2. Incorrect Touchstone Format

There are several common problems you might encounter when using Touchstone data:

- Having all of the data for one line. (The N-port matrix on one line of the data file for files bigger than 4 ports.) Remember that there is a maximum of 8 entries per line, so any file with more than 4 ports gets more complex because new lines are required after 8 entries are added. See "Touchstone File Format " for an example of a 5-port data file with line wrapping. If your data is not line-wrapped this way, you can use the raw data format described in "Raw Data File Format ", including a technique to convert data to properly formatted Touchstone files.

- Improper line wraps in the data file (Touchstone v1.0 format only). There are many situations where this might occur. The solution is to use the raw data format described in "Raw Data File Format ".
- Duplicate frequencies in the data file. Duplicate frequencies produce the following error: "Problem with file format: Error reading line <x>: expecting 5 entries per line for noise data". Upon finding a duplicate frequency, the program thinks it has entered the noise data section of the file.
- Having derived data (such as common mode rejection ratio) calculated from the raw network data. Some Vector Network Analyzers export Touchstone data files with derived data appended to each line. Contact Cadence AWR Support for scripts that help clean up this data.

3.3.3. N-Port Touchstone Files from Many 2-port Files

You can generate one N-port Touchstone file from many M-port Touchstone files; where N > M (typically, M=2 and N=3 or 4). Choose **Scripts > Data > Combine_S_Params** to run a Visual Basic script that performs this automatically.

3.3.4. Extrapolation Problems (Specifically at DC)

When Touchstone, raw data, and MDIF files are used as subcircuits in a larger circuit simulation, problems arise when the simulation occurs at frequencies outside of the range of the data files frequency range. In this case, the software must extrapolate a response for the data file from the existing data.

Extrapolating to DC can cause common problems such as current flowing through blocking capacitors or transistors not biasing up properly. One method to check for problems is to turn on both current and voltage DC annotations so you can see these simulation values on the schematic. After identifying a problem, there are several things you can do to fix it:

1. Change the interpolation/extrapolation options. You can access and change these options for the entire project by choosing **Options > Project Options** and clicking the **Interpolation/Passivity** tab. You can also access and change these options for a single data file by selecting the data file under the **Data Files** node in the Project Browser, right-clicking and choosing **Options**, and then clicking the **Interpolation/Passivity** tab. You can try changing the interpolation method or the coordinate system. See "Options - Data File Dialog Box: Interpolation/Passivity Tab " for more information on the options in this dialog box.
2. Edit the data file directly and add the proper entries for DC. To do so you must know the proper entries in your files. This is more difficult for MDIF files because you need to edit each block of data.
3. Place your data file in a schematic and then use large inductors and capacitors to define the proper DC paths (use a capacitor to block DC and inductors between ports where DC current should flow). You can now use the schematic as a subcircuit, or export it as an output file in Touchstone format to generate a new Touchstone format with proper entries at DC. Be careful using the schematic with large capacitors and inductors when using transient simulations, as these components introduce very large time constants resulting in the need for many cycles to get to steady state.

Another problem is the behavior at the harmonics of the fundamental. For example, this can occur if you have a 2 to 3 GHz amplifier and data for some capacitors from 1 to 4 GHz, and you want to run harmonic balance analysis to get the compression characteristics of the amplifier. You are running 5 harmonics in the harmonic balance simulation, so the simulation needs to know the behavior of those caps at 15 GHz (3 GHz x 5 harmonics). 15 GHz is significantly outside the range of the data file, so the extrapolated data is most likely not accurate.

3.3.5. Noise for Data Files

When you use Touchstone, raw data, and MDIF files with noise simulation, the program first determines if the data file is passive. If passive, the noise can be computed from the network parameters. If not passive, the data file expects to find noise parameters in the data file. Sometimes, data for passive structures can be slightly active (due to calibration errors or EM simulator numerical problems).

If this problem occurs, you can force the data file to be treated as passive for noise simulation. To do so, right-click the file in the Project Browser under the **Data Files** node, choose **Options**, click the **Interpolation/Passivity** tab, and then select the **Consider Passive for Noise Simulation** check box. For more information on the options in this dialog box, see "Options - Data File Dialog Box: Interpolation/Passivity Tab ".

3.3.6. Grounding Types

When you use S-parameters in a schematic, they are inserted as a subcircuit. You have three options for grounding types: normal, explicit ground node, and balanced ports. You can find these options on the Element Options: SUBCKT - Properties dialog box **Ground** tab (right-click the S-parameter subcircuit and choose **Properties**.) **Explicit ground node** exposes the ground node so it is accessible in the schematic. **Balanced ports** adds a local "ground" port to each port of the S-parameter file.

You can view an exposed ground node as a local ground for the S-parameter file. It is important to understand that the same ground is used for all ports in the S-parameter file. This implies that, physically, the structure is electrically small, or has a very good (perfect) internal grounding system connecting the ports. Normally, exposing the ground node is used for transistor data, where a common ground node in the measurement is being exposed.

Balanced ports extend the exposed ground node concept by creating a local ground node for each port. Conceptually this is similar to attaching an ideal 1:1 transformer to each port, and using the exterior coil to create a local ground reference. It is possible to misuse this concept and obtain physically meaningless results.

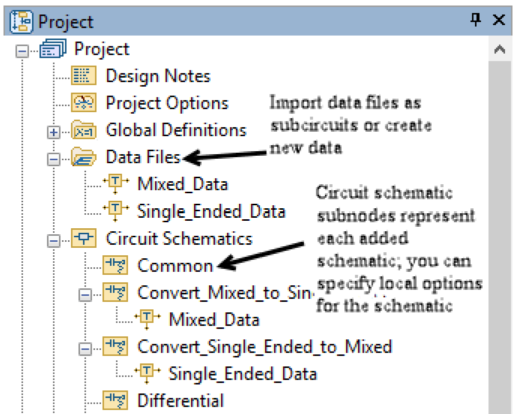
To learn more about the different grounding types choose **File > Open Example** and search for "ground_node". See the Design Notes for the example for more information about different grounding types.

Chapter 4. Schematics and System Diagrams

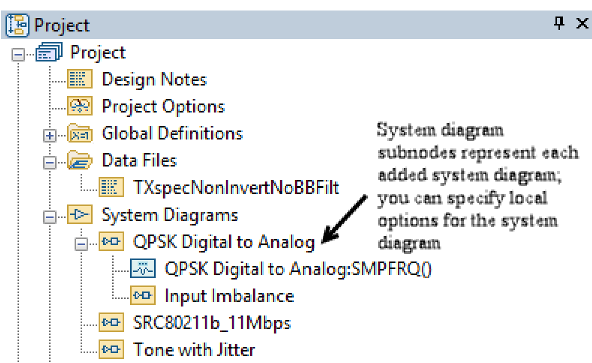
Schematics are graphical representations of circuits composed in the Cadence® Microwave Office® software. An Microwave Office project can include multiple schematics. System diagrams are representations of complete communication systems composed in the Cadence Visual System Simulator™ (VSS) communications and radar systems software. An VSS software project can include multiple system diagrams.

4.1. Schematics and System Diagrams in the Project Browser

The **Circuit Schematics** node in the Project Browser contains a subnode for each schematic that you create or import into the Cadence AWR Design Environment® platform for that project. The following figure shows the **Circuit Schematics** node and its subnodes.



The **System Diagrams** node in the Project Browser contains a subnode for each system diagram that you create or import into the AWR Design Environment platform for that project. The following figure shows the **System Diagrams** node and its subnodes.



4.2. Creating, Importing, or Linking to Schematics

To create a new schematic:

1. Right-click **Circuit Schematics** in the Project Browser and choose **New Schematic**, or choose **Project > Add Schematic > New Schematic**.

   The New Schematic dialog box displays.

2. Enter a name for the schematic. Click **Create**. An empty schematic window opens in the workspace, and the Project Browser displays the new schematic and its subnodes under **Circuit Schematics**. For information on how to add elements to the new schematic, see "Adding Elements Using the Elements Browser".

If there are multiple process libraries (PDKs) in the project, the New Schematic dialog box lists the PDKs you can choose to associate with the new schematic. The Global Definitions document from the PDK is automatically set for the Equations option of the new schematic, and the LPF from the PDK is automatically set for the Schematic option of the new schematic. For information on how to add PDKs to a project, see "Working With Foundry Libraries".

To import or link to an existing schematic:

1. Right-click **Circuit Schematics** in the Project Browser and choose **Import Schematic**, or choose **Project > Add Schematic > Import Schematic**.

   The Browse for File dialog box displays.

2. Locate the desired schematic (imported schematics have a **.sch* file extension) and click **Open** to copy the file and make it part of the project. As with creating a new schematic, a schematic window opens in the workspace, and the Project Browser displays the imported schematic and its subnodes under **Circuit Schematics**.

   Alternatively, you can access a schematic without copying it into the project. To link to a schematic, right-click **Circuit Schematics** in the Project Browser, and choose **Link to Schematic**. The Browse for File dialog box displays. Locate the desired schematic and click **Open** to make the file part of the project. A schematic window opens in the workspace, and the Project Browser displays the linked schematic and its subnodes under **Circuit Schematics**.

   NOTE: When you link to a schematic, that file must always be available for the project to read.

4.3. Creating, Importing, or Linking to System Diagrams

To create a new system diagram:

1. Right-click **System Diagrams** in the Project Browser and choose **New System Diagram**, or choose **Project > Add System Diagram > New System Diagram**.

   The New System Diagram dialog box displays.

2. Enter a name for the system diagram and click **Create**. An empty system diagram window opens in the workspace, and the Project Browser displays the new system diagram and its subnodes under **System Diagrams**. For information on how to add system blocks to the new system diagram, see "Adding System Blocks Using the Elements Browser".

To import or link to an existing system diagram:

1. Right-click **System Diagrams** in the Project Browser and choose **Import System Diagram**, or choose **Project > Add System Diagram > Import System Diagram**.

   The Browse for File dialog box displays.

2. Locate the desired system diagram (imported system diagrams have a **.sys* file extension) and click **Open** to copy the file and make it part of the project. As with creating a new system diagram, a system diagram window opens in the workspace, and the Project Browser displays the imported system diagram and its subnodes under **System Diagrams**.

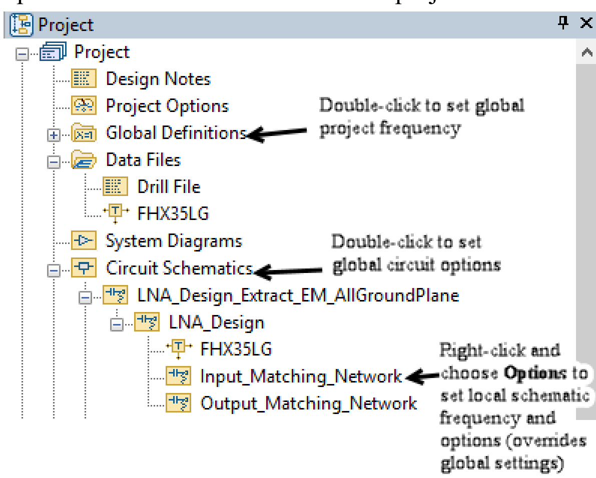
   Alternatively, you may want to access a system diagram without copying it into the project. To link to a system diagram, right-click **System Diagrams** in the Project Browser, and choose **Link to System Diagram**. The Browse for File dialog box displays. Locate the desired system diagram and click **Open** to make the file part of the project. A system diagram window opens in the workspace, and the Project Browser displays the linked system diagram and its subnodes under **System Diagrams**.

NOTE: When you link to a system diagram, that file must always be available for the project to read.

4.4. Specifying Schematic and System Diagram Options

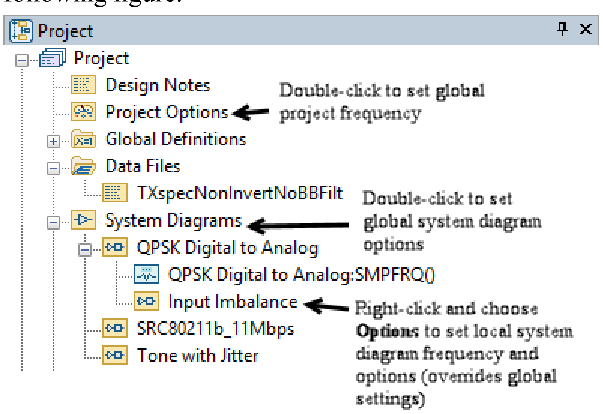
Schematic options include settings that control how the harmonic balance simulator performs its calculations and what type of solver is applied to linear simulations, as well as simulation frequency.

You can configure schematic options for a particular schematic via the schematic's subnodes, or you can use the default options set for all circuits within the project. These choices are shown in the following figure.



System diagram options include simulation control, RF, and RF Inspector settings, as well as simulation frequency.

You can configure system diagram options for a particular system diagram via the system diagram's subnodes, or you can use the default options set for all system diagrams contained within the project. These choices are shown in the following figure.



4.4.1. Configuring Global Circuit Options

To configure global circuit options for schematics:

1. Do one of the following:
   - Choose **Options > Default Circuit Options**, or
   - Double-click **Circuit Schematics** in the Project Browser.

   The Circuit Options dialog box displays.
2. Make your modifications, and click **OK**. If you don't configure global circuit options, the default global circuit options are used.
3. Follow the instructions in "Configuring Global Project Frequency" to configure global frequencies. If you don't configure global frequencies, the default global frequency shown in the figure in "Project Options Dialog Box: Frequencies Tab " is used.

To configure global circuit options for system diagrams:

1. Do one of the following:
   - Choose **Options > Default System Options**, or
   - Double-click **System Diagrams** in the Project Browser.

   The System Simulator Options dialog box displays.
2. Make your modifications, and click **OK**. If you don't configure global system dialog options, the default global system dialog options are used.
3. Follow the instructions in "Configuring Global Project Frequency" to configure global frequencies. If you don't configure global frequencies, the default global frequency shown in the figure in "Project Options Dialog Box: Frequencies Tab " is used.

4.4.2. Configuring Local Schematic or System Diagram Options and Frequency

To configure local schematic options:

1. Double-click **Circuit Schematics** in the Project Browser. The Circuit Options dialog box displays.
2. Specify circuit options.
3. Click **OK**.

To configure local system dialog options:

1. Double-click **System Diagrams** in the Project Browser. The System Simulator Options dialog box displays.
2. Specify system diagram options.
3. Click **OK**.

To configure local simulation frequency:

1. Right-click the desired schematic or system diagram and choose **Options**. The Options dialog box displays.
2. Clear the **Use project defaults** check box, and then specify the desired local simulation frequency by setting the frequency range **Start**, **Stop**, and **Step** values. For schematics, see "Options Dialog Box: Frequencies Tab", and for system diagrams, see "System Simulator Options Dialog Box: RF Frequencies Tab" for more information.

To define multiple frequency sweeps within a schematic you can add the Swept Frequency control (SWPFRQ) in the Elements Browser **Simulation Control** category by including it in a schematic. See "Frequency Sweep Control" for more information.

3. Click **OK**.

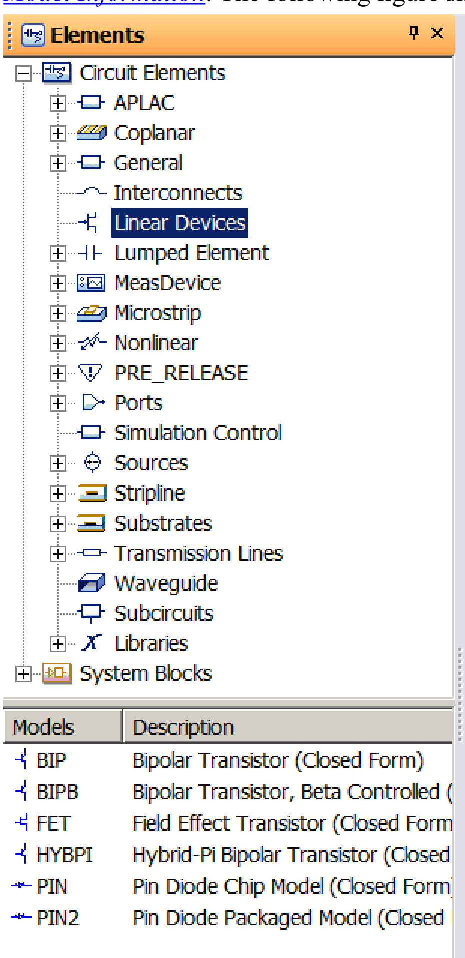
4.5. Working with Elements on a Schematic

This section includes information on how to add elements to a schematic, how to manipulate elements on a schematic, and how to edit or use variables and equations for element parameters. Information about both Schematic and Layout Views is also included.

4.5.1. Adding Elements Using the Elements Browser

The Elements Browser allows you to browse through a comprehensive database of hierarchical groups of electrical entities such as lumped elements or microstrips, and select the desired model to include in your schematics.

For a complete description of all the elements in the Elements Browser, see the AWR Microwave Office Element Catalog. For a description of the XML Libraries, see Appendix A, *Component Libraries*. For special notes regarding microstrip iCells, linear models for transmission line systems, and EM-based discontinuity models, see Appendix A, *Supplemental Model Information*. The following figure shows the Elements Browser.



To add an element to a schematic:

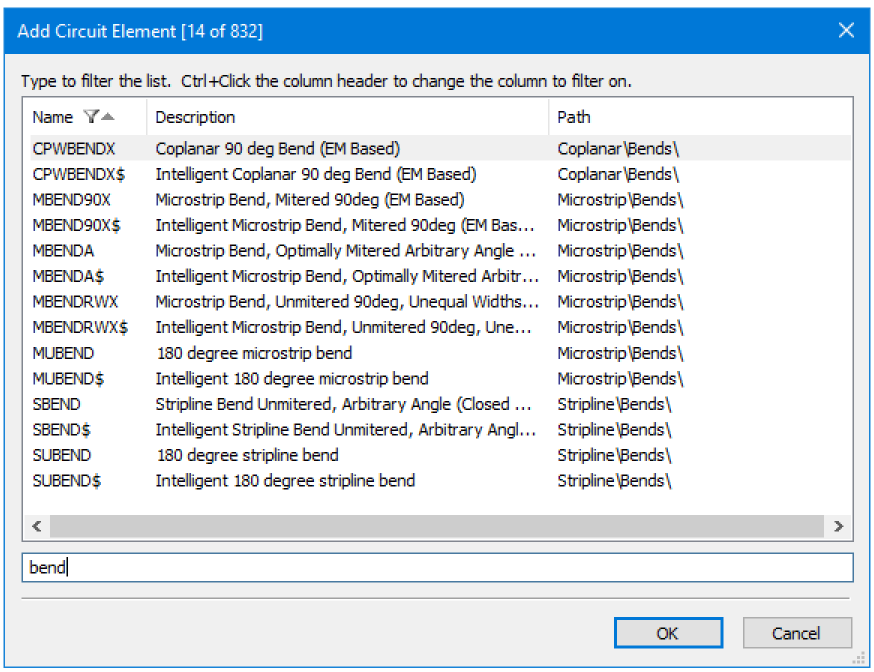
1. Click the **Elements** tab on the main window to display the Element Browser.
2. If necessary, double-click **Circuit Elements** to open the group. Click the **+** and **-** symbols to expand and contract the groups of elements, and click the desired subgroup, such as **Active** or **Passive**. The available models display in the lower window pane.
3. To place a desired model, click and drag it into the schematic window, release the mouse button, position the element, and click to place it. When positioning the element right-click to rotate it, **Shift**+right-click to flip it horizontally, and **Ctrl**+right-click to flip it vertically.

You can also copy element information to another instance of the Microwave Office software by selecting the element and choosing **Edit > Copy**. In the Project Browser of the second application, select the target and then choose **Edit > Paste**.

To add a shape to a schematic, choose the desired shape from the **Draw** menu and then click in the schematic window to begin drawing the shape. For information on drawing shapes, see [“Schematic/EM Layout Draw Tools ”](#).

4.5.2. Adding Elements Using the Add Element Command

The Add Element command allows you to add elements from a list dialog box that supports filtering by element name, description, or library path. In a Schematic View, choose **Draw > More Elements**, press **Ctrl + L**, or click the **Element** button on the Schematic Design toolbar to display the Add Circuit Element dialog box.



The Add Circuit Element dialog box provides several ways to filter elements:

- To filter the list by name, **Ctrl**-click the **Name** column header and begin typing an element name in the text box at the bottom of the dialog box.

  The element list is filtered to display only those elements that match your input.

- To filter the list by description, **Ctrl**-click the **Description** column header and begin typing a description in the text box at the bottom of the dialog box.

The element list is filtered to display only those elements whose description includes the typed text.

You can type more than one match word. For example, typing **micro bend** when matching on the **Description** column displays only microstrip bend elements.

- To filter the list by path, **Ctrl**-click the **Path** column header and begin typing a directory path in the text box at the bottom of the dialog box.

  The element list is filtered to display only those elements whose directory path includes the typed text.

Click a column header to sort by that column. Click again to reverse the sort order for that column.

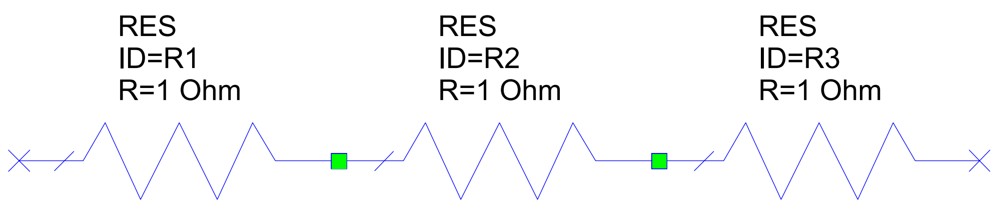
After filtering to find the desired element, select the element and click **OK** to add the element to the active schematic.

If you work with Process Design Kits (PDKs), this dialog box includes elements available in the PDK you are using. You can easily filter to show only parts from that PDK. Typically, all PDK models begin with the foundry name, so you can filter by foundry name after selecting the **Name** column. You can also filter by the **Path** column and use **Libraries** as the filter.

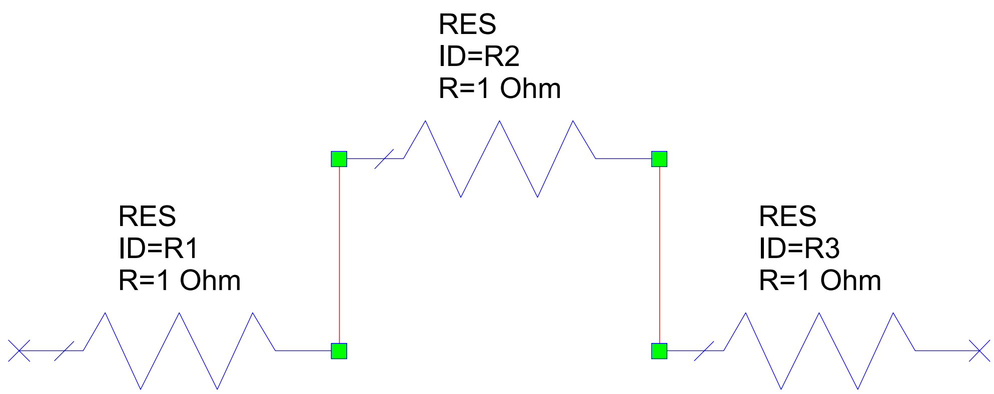
Unless you install a local copy of the web libraries (available as a download from the Cadence website), this dialog box does not display parts available in the **Libraries > *AWR web site** category of the Elements Browser.

4.5.3. Moving, Rotating, Flipping, and Mirroring Elements

To move an element in the schematic, click the element then drag it to a new position, as shown in the following figures.



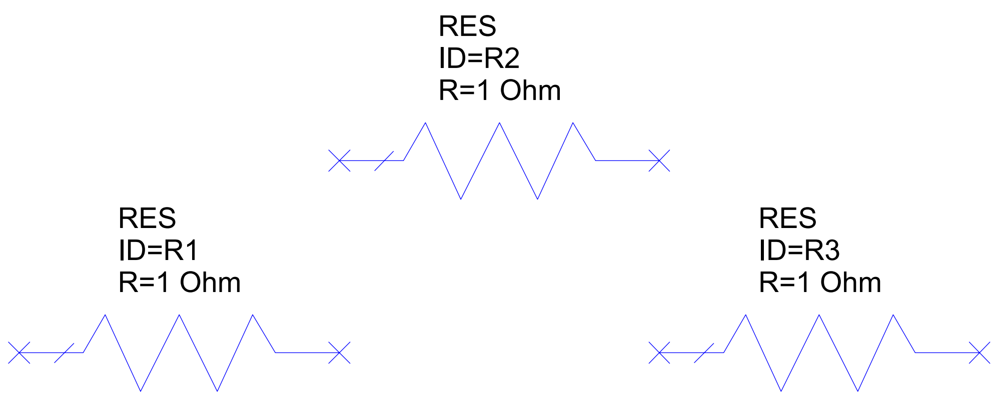
Connection wires are automatically added to keep the element connected, as shown in the following figure. If the element already has connecting wires, the wires stretch. If wires are stretched such that the wire segments fall on top of other nodes in the circuit, the wires connect to those nodes also (they exhibit "sticky" behavior).



If you press the **Ctrl** key while moving the element, no connecting wires are added, as shown in following figure. If there are already connecting wires on the element, pressing the **Ctrl** key while moving the element removes the wire connections.

If you press the **Shift** key while moving the element, the movement is restricted to only horizontal or vertical from the original location.

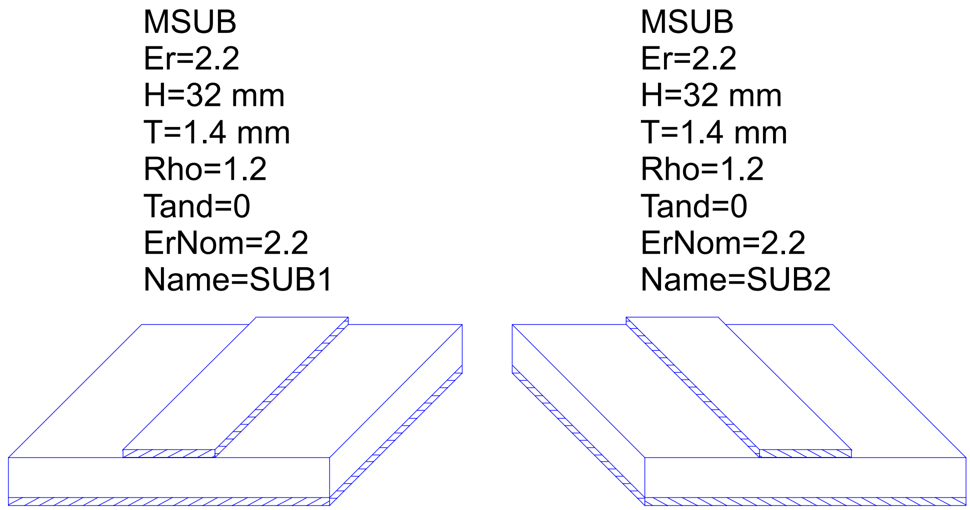
You can rotate or flip elements by selecting the element, right-clicking, and choosing **Rotate** or **Flip**. When elements are rotated or flipped, the wire connections are automatically broken unless the node points of the rotated or flipped element end up at the same location as the original element, as shown in the following figure. For instance, a two-node element can be reversed by starting the rotate command, clicking on the midpoint between the two nodes, and then dragging the mouse to rotate the element 180-degrees. The rotated element's nodes then fall on the same points as the original node positions, and any connecting wires remain connected.



4.5.3.1. Element Mirroring

You can also create a mirrored image of an element in a schematic.

To access Mirroring, in a schematic window select the desired element and choose **Edit > Mirror**. The cursor changes to reflect the mirroring operation. Click in the schematic to position the new element. The following figure shows a mirroring operation.



4.5.4. Editing Element Parameter Values

To edit an element's parameter values:

1. Double-click the element graphic in the schematic window. The Element Options dialog box displays. For more information about this dialog box, see the screens starting with "Element Options Dialog Box: Parameters Tab".
2. Make the necessary parameter modifications, and click **OK**.

You can also edit parameter values directly on a schematic by double-clicking the parameter value in the schematic window. An edit box displays to allow you to modify the value. Press the **Tab** key to quickly move to the next parameter entry. Press **Shift+Tab** to move to the previous parameter.

You can use the following standard unit modifiers to simplify entry of model parameters:

f	1e-15
p	1e-12
n	1e-9
u	1e-6
m	1e-3
c	1e-2
mil	25.4e-6
k	1e3
meg	1e6
g	1e9
t	1e12

For example, if you are working in base units you can enter "1p" instead of "1e-12" for a capacitor. You can also use modifiers in equations.

These modifiers follow SPICE rules; they are not case sensitive, they must follow the number directly without a space in between, and any characters directly following the modifier are ignored. Note that the suffix "d" (deci) is not supported since it is reserved for use as an alternative to "e" in scientific notation.

4.5.4.1. Selecting Multiple Elements

There are various ways to select multiple elements. All elements in your current selection group display with selection boxes around them and their parameter text outlined.

- Press the **Shift** key while individually clicking on elements to add them to a selection group. Click on them again to remove them from the selection group.
- Click and drag the mouse to define a selection area, then release the mouse button. All elements *completely* enclosed in this area are selected.
- **Shift**+click and drag the mouse to define a selection area, then release the mouse button. All elements completely or partially enclosed in this area are selected.
- **Shift+Ctrl**click on elements to cycle through elements that overlap. By default, the smallest object is selected first and then larger items are selected as you cycle through them.
- With a schematic window active, choose **Edit > Select Tool** to display the Selection Tool dialog box and select all items that match certain criteria. The dialog box displays the number of items found. When you close the dialog box the items are still selected so you can then edit them. See “Element Selection Tool Dialog Box” for more information.

4.5.4.2. Editing Multiple Elements

To edit multiple elements simultaneously, select multiple elements, right-click one of them and choose **Properties** or choose **Edit > Properties**. The Element Options: Multiple Element Type Properties dialog box displays to allow you to edit any common element parameters. If the parameter values are identical, the value is displayed in the dialog box. If the parameter values are different, the value displays as "***".

4.5.4.3. Editing Element IDs

The first parameter for each element is the ID of the element. You can edit the ID to make it more meaningful. The following special characters are not allowed in element IDs:

- (
-)
- , (comma)
- =
- \
- " (double quote)
- ' (single quote)
- ` (back tick)
- (space)

4.5.5. Using Variables and Equations for Parameter Values

Microwave Office software allows you to define variables and equations to express parameter values within schematics. To assign a parameter value to a variable, create the required variables and equations as described in “*Variables and Equations*”, then edit the parameter value as described previously, specifying the variable name as its new value.

Another type of syntax supported by parameters allows you to refer to other parameters of the same element, different elements connected to a node of the element, or elements specified by the element name and ID. The following table shows these three syntax forms.

Parameter Syntax	Description
P1=P2@	P1 uses the value of parameter P2 from this element
P=P@1	P uses the value of parameter P from an element connected to node 1 of this element
W=W@MLIN.TL1	W uses the value of parameter W from an MLIN element whose ID=TL1

4.5.6. Using Elements with Model Blocks

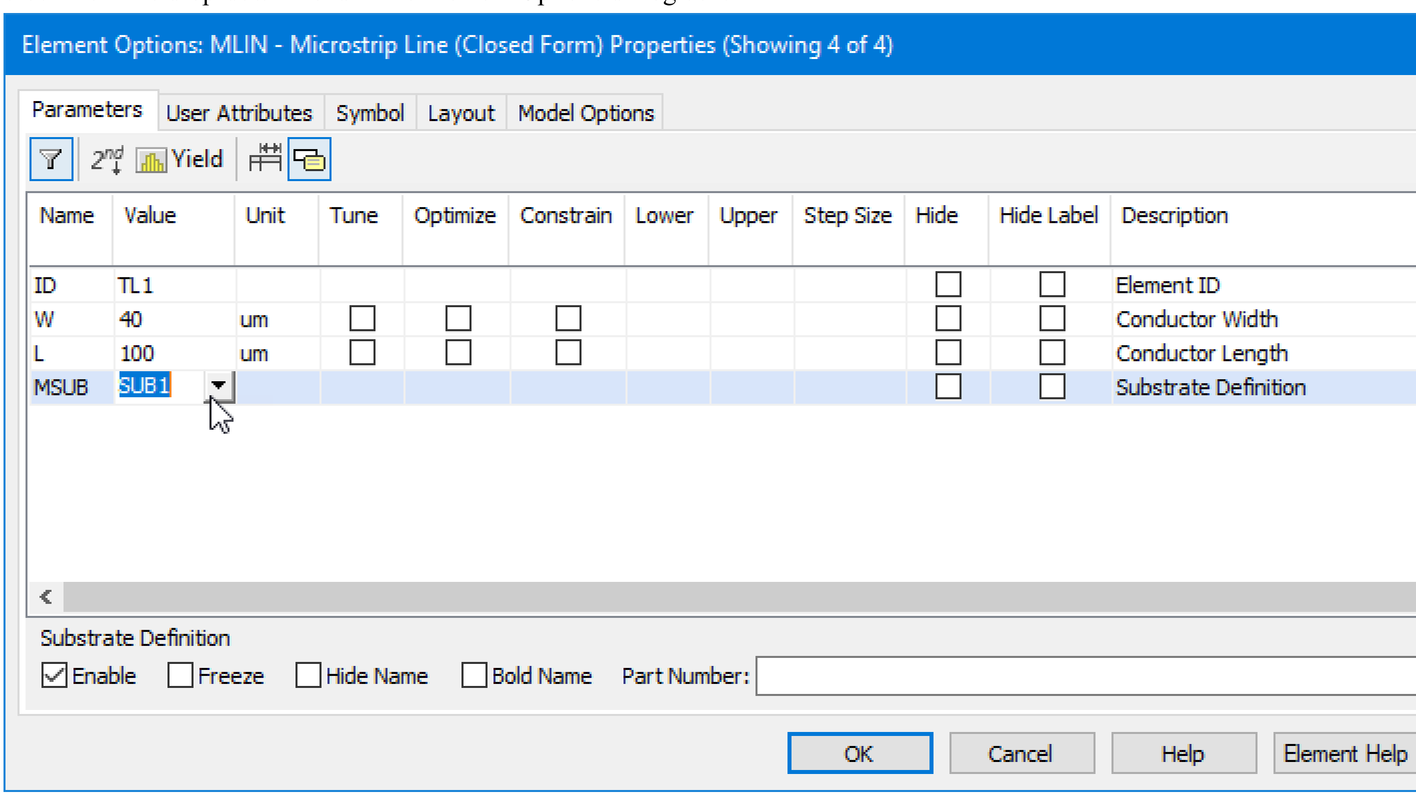
Some elements in the AWR Design Environment platform have parameter settings for model blocks. A common example is for transmission lines. The individual elements define the transmission line geometry and then have a parameter for the substrate element used to define the substrate parameters for this transmission line.

The type of model block needed is the parameter name listed for the element. For example, an MLIN model has an MSUB parameter, meaning that an MSUB element is required for this element to be used in simulation.

To add a model block to a schematic, right-click an element that requires a model block and choose **Add Model Block**.

4.5.6.1. Model Block Concerns

By default, the model block element entries are blank. You can identify specific model blocks by selecting the model block from the drop-down menu in the Element Options dialog box.



If the parameter is empty, the AWR Design Environment platform uses any model block found after searching the following:

- the same schematic as the element using the model block
- the Global Definitions

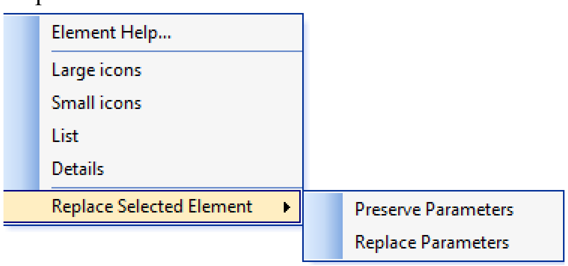
If more than one model block is defined at one location when the model block parameter is empty, the AWR Design Environment software issues an error and simulation stops. You can double-click the displayed error message to go to the problem model.

4.5.7. Swapping Elements

The Swap Elements command allows swapping one or multiple elements, swapping with an element that has a different number of nodes from the original, and swapping with elements from XML libraries.

To swap one or more elements, select the element(s) in the schematic, right-click and choose **Swap Elements**. In the Swap Elements dialog box, choose the element with which you want to replace the selected element, and specify preservation or replacement of the symbol and/or electrical parameters of the swapped element.

Alternatively, you can select the element(s) in the schematic, and in the Elements Browser, right-click on the element with which you want to replace the selected element(s). Choose **Replace Selected Element > Preserve Parameters** or **Replace Selected Element > Replace Parameters** to replace the selected element with the specified element and preserve or replace its parameters.



You can also edit a schematic element by double-clicking it and changing its name. Changing the name of the element is equivalent to swapping with parameter preservation.

4.5.8. Restricted Object Selection

Restricted object selection is added in Schematic Views to prevent objects from being selected. To use this feature, right-click in the schematic window, choose **Restrict Selection** and then select the item types to restrict. Selecting an item type prohibits it from being selected in the schematic. If you find you cannot select certain items in a schematic, you should verify that they have not been restricted from selection. See "Restrict Selection (Schematics) Dialog Box" for more information.

4.5.9. Viewing the Layout for a Schematic

The Schematic View and the Layout View are two views of a single intelligent database that manages the connectivity between the circuit components. To view the layout for a specific element in a schematic, select the element in the schematic window, right-click and choose **Select in Layout**. The layout window displays the schematic layout with the specified element's layout or artwork cell highlighted (if it has an assigned cell). For more information about these two views see "The Layout as Another View of the Schematic Database". For more information on interaction between the schematic and the layout see "Schematic and Schematic Layout Interaction ".

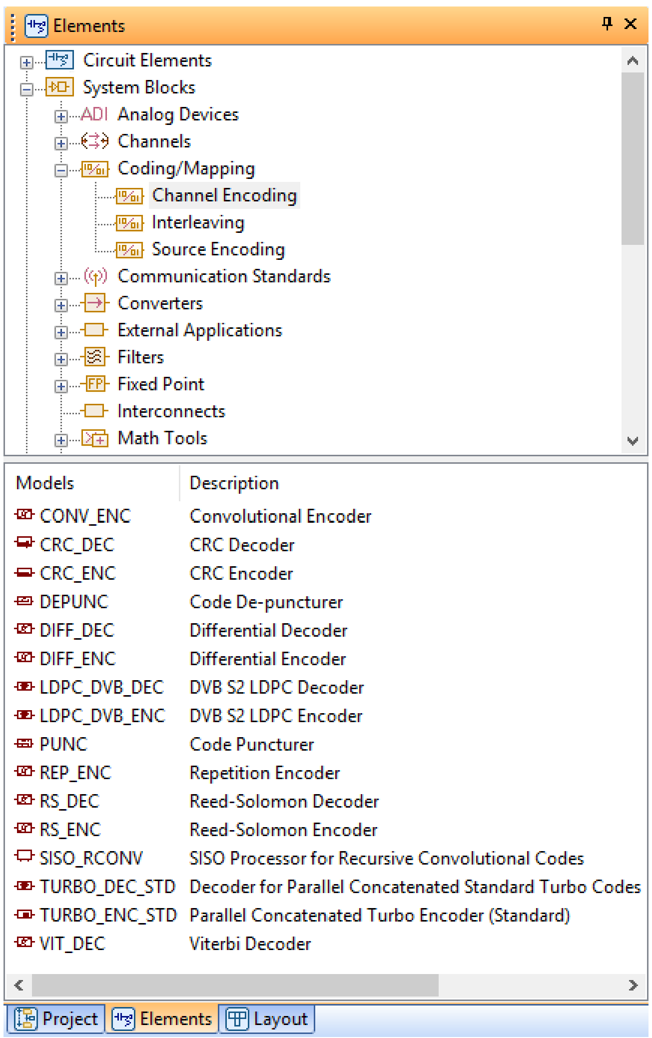
4.6. Working with System Blocks on a System Diagram

This section includes information on how to add system blocks to a system diagram, how to manipulate system blocks on a system diagram, and how to edit or use variables and equations for system block parameters. Information about restricting object selection in a System Diagram View is also included.

4.6.1. Adding System Blocks Using the Elements Browser

The Elements Browser allows you to browse through a comprehensive database of system blocks such as analog devices or converters, and select the desired system block to include in your system diagram.

For a complete description of all the system blocks in the Elements Browser, see the [AWR Visual System Simulator System Block Catalog](). For a description of the XML Libraries, see [Appendix A, *Component Libraries*](). The following figure shows the Elements Browser.



To add a system block to a system diagram:

1. Click the **Elements** tab on the main window to display the Element Browser.

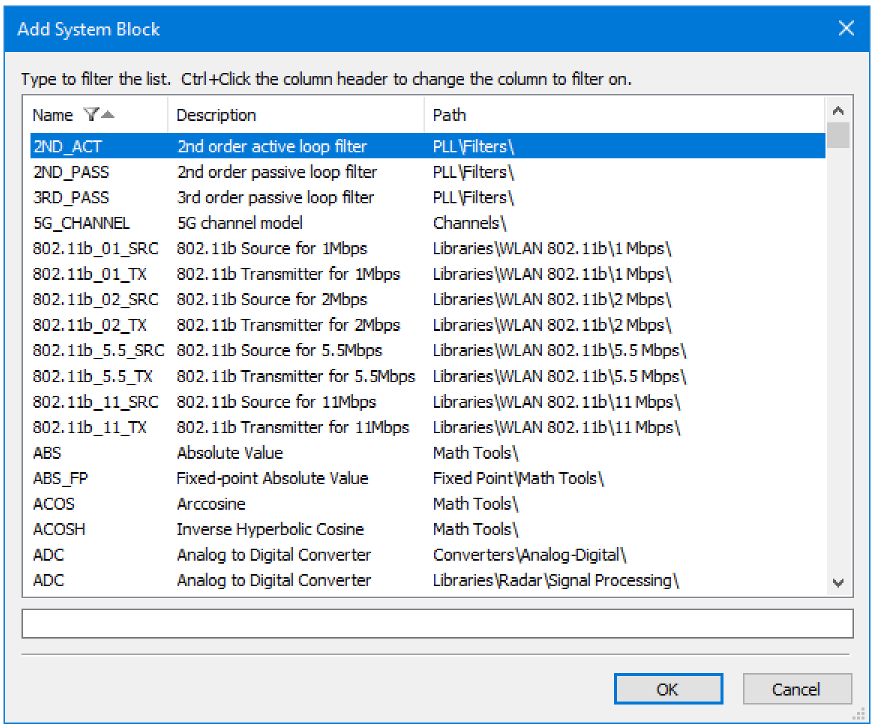
2. If necessary, double-click **System Blocks** to open the group. Click the **+** and **-** symbols to expand and contract the groups of system blocks, and click the desired subgroup, such as **Channel Encoding** or **Analog-Digital**. The available blocks display in the lower window pane.
3. To place a desired block, click and drag it into the system diagram window, release the mouse button, position the block, and click to place it. When positioning the block right-click to rotate it, **Shift**+right-click to flip it horizontally, and **Ctrl**+right-click to flip it vertically.

You can also copy block information to another instance of the VSS software by selecting the block and choosing **Edit > Copy**. In the Project Browser of the second application, select the target and then choose **Edit > Paste**.

To add a shape to a system block, choose the desired shape from the **Draw** menu and then click in the system diagram window to begin drawing the shape. For information on drawing shapes, see “Schematic/EM Layout Draw Tools ”.

4.6.2. Adding System Blocks Using the Add Element Command

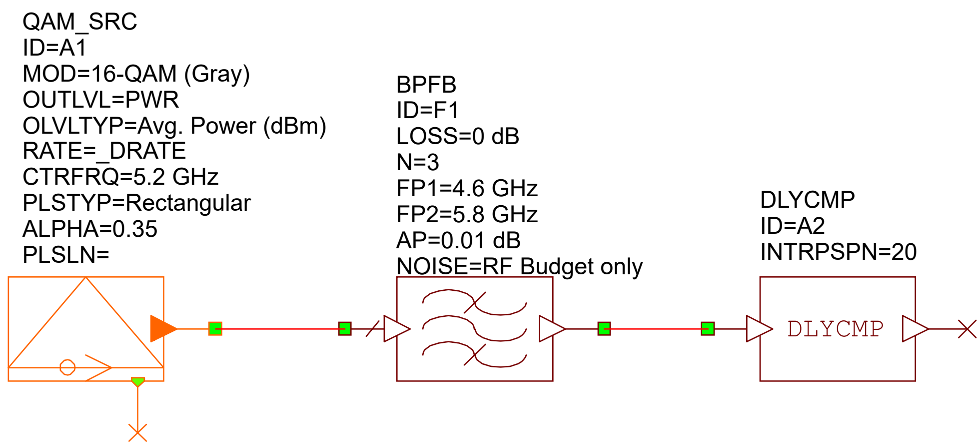
The Add Element command allows you to add system blocks from a list dialog box that supports filtering by system block name, description, or library path. In a System Diagram View, choose **Draw > More Elements**, press **Ctrl + L**, or click the **Element** button on the System Design toolbar to display the Add System Block Element dialog box.



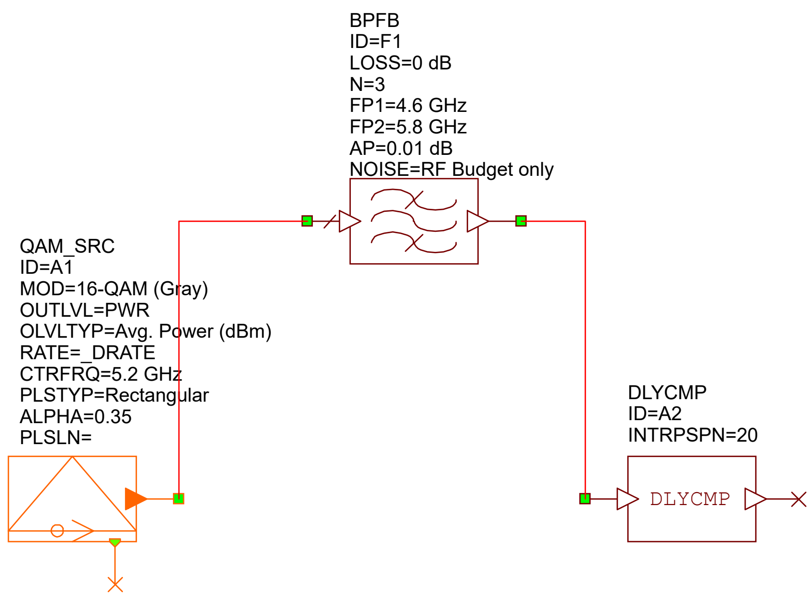
See the “Adding Elements Using the Add Element Command” for details on sorting and filtering items in this dialog box.

4.6.3. Moving, Rotating, Flipping, and Mirroring System Blocks

To move a system block in the system diagram, click on the block, then drag the block to a new position, as shown in the following figures.



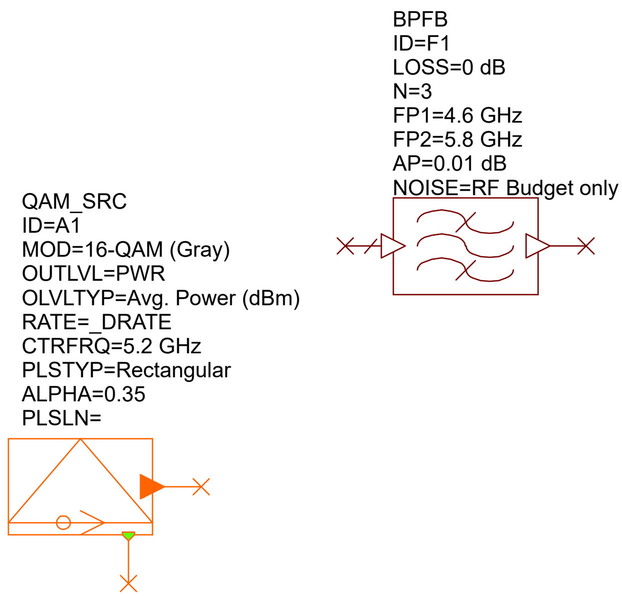
Connection wires are automatically added to keep the block connected, as shown in the following figure. If the block already has connecting wires, the wires stretch. If wires are stretched such that the wire segments fall on top of other nodes in the system diagram, the wires connect to those nodes also (they exhibit "sticky" behavior).



If you press the **Ctrl** key while moving the block, no connecting wires are added, as shown in the following figure. If there are already connecting wires on the block, pressing the **Ctrl** key while moving the block removes the wire connections.

If you press the **Shift** key while moving the block, the movement is restricted to only horizontal or vertical from the original location.

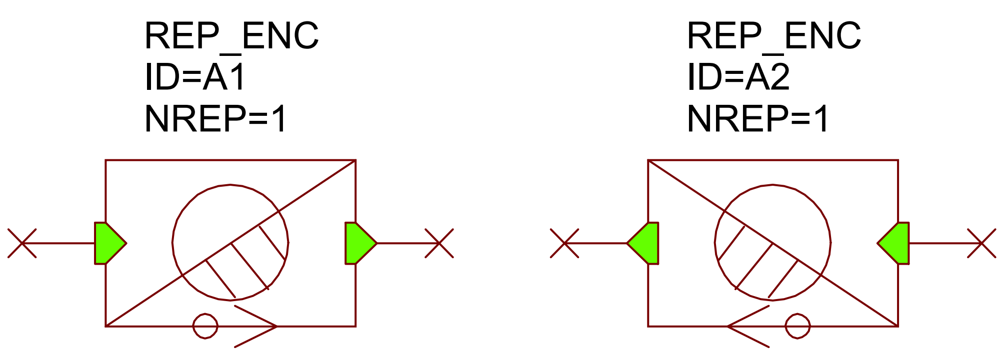
You can rotate or flip blocks by selecting the block, right-clicking, and choosing **Rotate** or **Flip**. When blocks are rotated or flipped, the wire connections are automatically broken unless the node points of the rotated or flipped block end up at the same location as the original block, as shown in the following figure. For instance, a two-node block can be reversed by starting the rotate command, clicking on the midpoint between the two nodes, and then dragging the mouse to rotate the block 180-degrees. The rotated block's nodes then fall on the same points as the original node positions, and any connecting wires remain connected.



4.6.3.1. System Block Mirroring

You can also create a mirrored image of a system block in a system diagram.

To access Mirroring, in a system diagram window select the desired block and choose **Edit > Mirror**. The cursor changes to reflect the mirroring operation. Click in the system diagram to position the new block. The following figure shows a mirroring operation.



4.6.4. Editing System Block Parameter Values

To edit a system block's parameter values:

1. Double-click the system block graphic in the system diagram window. The Element Options dialog box displays. For more information about this dialog box, see the screens starting with "Element Options Dialog Box: Parameters Tab".
2. Make the necessary parameter modifications, and click **OK**.

You can also edit parameter values directly on a system diagram by double-clicking the parameter value in the system diagram window. An edit box displays to allow you to modify the value. Press the **Tab** key to move to the next parameter entry.

You can use the following standard unit modifiers to simplify entry of model parameters:

f	1e-15
p	1e-12
n	1e-9
u	1e-6
m	1e-3
c	1e-2
d	1e-1
mil	25.4e-6
k	1e3
meg	1e6
g	1e9
t	1e12

For example, if you are working in base units you can enter "1p" instead of "1e-12". You can also use modifiers in equations.

These modifiers are not case sensitive, they must follow the number directly without a space in between, and any characters directly following the modifier are ignored.

4.6.4.1. Selecting Multiple System Blocks

There are various ways to select multiple system blocks. All blocks in your current selection group display with selection boxes around them and their parameter text outlined.

- Press the **Shift** key while individually clicking on blocks to add them to a selection group. Click on them again to remove them from the selection group.
- Click and drag the mouse to define a selection area, then release the mouse button. All blocks *completely* enclosed in this area are selected.
- **Shift**-click and drag the mouse to define a selection area, then release the mouse button. All blocks completely or partially enclosed in this area are selected.
- With a system diagram window active, choose **Edit > Select Tool** to display the Selection Tool dialog box and select all items that match certain criteria. The dialog box displays the number of items found. When you close the dialog box the items are still selected so you can then edit them. See "Element Selection Tool Dialog Box" for more information.

4.6.4.2. Editing Multiple System Blocks

To edit multiple blocks simultaneously, select multiple blocks, right-click one of them and choose **Properties** or choose **Edit > Properties**. The Element Options: Multiple Element Type Properties dialog box displays to allow you to edit any common block parameters. If the parameter values are identical, the value is displayed in the dialog box. If the parameter values are different, the value displays as "***".

4.6.4.3. Editing System Block IDs

The first parameter for each element is the ID of the element. You can edit the ID to make it more meaningful. The following special characters are not allowed in element IDs:

- (
-)
- , (comma)
- =
- \
- " (double quote)
- ' (single quote)
- ` (back tick)
- (space)

4.6.5. Using Variables and Equations for Parameter Values

The AWR Design Environment platform allows you to define variables and equations to express parameter values within system diagrams. To assign a parameter value to a variable, create the required variables and equations as described in "*Variables and Equations*", then edit the parameter value as described previously, specifying the variable name as its new value.

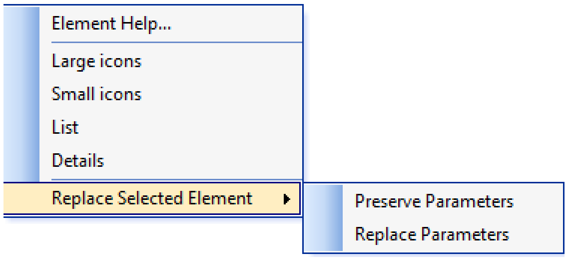
NOTE: VSS software uses base units (for example; Hz, seconds, Kelvin, or dBW) when you specify a variable or equation for a parameter value. Global units are used when you specify a numerical value.

4.6.6. Swapping System Blocks

The Swap Elements command allows swapping one or multiple blocks, swapping with a block that has a different number of nodes from the original, and swapping with blocks from XML libraries.

To swap one or more blocks, select the block(s) in the system diagram, right-click and choose **Swap Elements**. In the Swap Elements dialog box, choose the system block with which you want to replace the selected block, and specify preservation or replacement of the symbol and/or electrical parameters of the swapped block.

Alternatively, you can select the block(s) in the system diagram, and in the Elements Browser, right-click on the system block with which you want to replace the selected block(s). Choose **Replace Selected Element > Preserve Parameters** or **Replace Selected Element > Replace Parameters** to replace the selected block with the specified block and preserve or replace its parameters.



You can also edit a system block by double-clicking it and changing its name. Changing the name of the block is equivalent to swapping with parameter preservation.

4.6.7. Restricted Object Selection

Restricted object selection is added in system diagram views to prevent objects from being selected. To use this feature, right-click in the system diagram window, choose **Restrict Selection** and then select the item types to restrict. Selecting an item type prohibits it from being selected in the system diagram. If you find you cannot select certain items in a diagram, you should verify that they have not been restricted from selection. See “Restrict Selection (System Diagrams) Dialog Box” for more information.

4.7. Adding and Editing Ports

The Microwave Office program has two types of ports. The PORT element is a traditional microwave port that defines a port impedance used in simulation. There are several variations of this port to change how the impedance is defined. There are also ports that are large signal sources for nonlinear simulation. The PORT_NAME element is a special port that does not define a port impedance that is intended for use through hierarchy for simulation.

The VSS program also has two types of ports. Input ports (PORTDIN) are the entry point of data into a block, and receive data from an output port (PORTDOUT) of another block. When a simulation runs, data flows from the output port of one block to one or more input ports of other blocks connected to the output port.

4.7.1. Using PORTS

To add ports to a schematic or system diagram:

1. Click the **Ports** category in the Elements Browser.
2. For the Microwave Office program, click the desired port subgroup, such as **Signals**. The available models display in the lower window pane.
3. To place a desired port model into a schematic or system diagram, drag it into the window, position it, and click to place it.

Alternatively, in the Microwave Office program you can choose **Draw > Add Port** and in the VSS program you can choose **Draw > Add Input Port** or **Draw > Add Output Port**. You can also click the **Port** button on the toolbar, then add the entity to the schematic or system diagram.

To edit a port:

1. Double-click the port in the schematic or system diagram window.
2. In the Microwave Office program, click the **Port** tab. For more information about this dialog box tab, see “Element Options Dialog Box: Port Tab”.

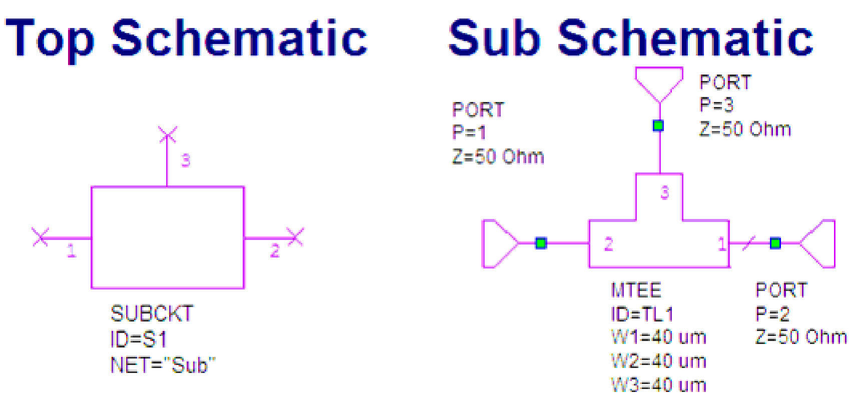
3. Make your selections and click **OK**.

Ports must be numbered sequentially from 1. The software increments any new ports added to a schematic or system diagram. Deleting ports or editing the port number can break this sequence.

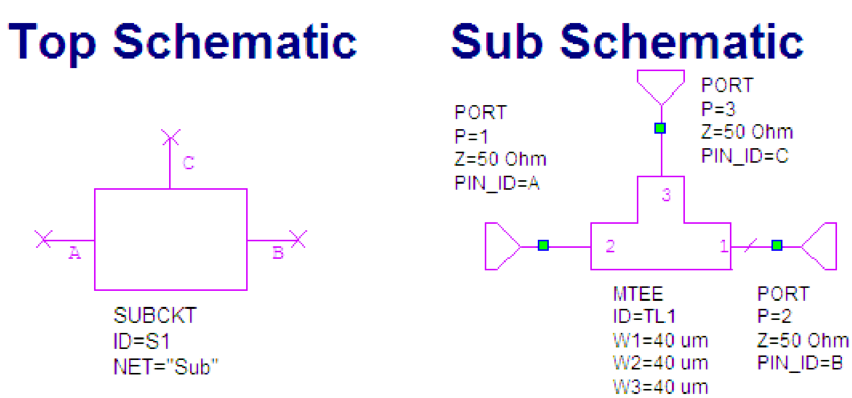
You cannot wire ports directly together or a simulation error occurs. If you must do so, add a 0 ohm resistor between the two ports.

4.7.1.1. PIN_ID and Hierarchy

Each Microwave Office port has a PIN_ID parameter that is used to identify the subcircuit pins by the name typed in, rather than the port number, if the schematic is used as a subcircuit. For example, the following figure shows a Microwave Office subcircuit using ports that don't use the PIN_ID parameter. Notice that the top level schematic that identifies the subcircuit connects with the numbers of the ports from the subcircuit.



If the PIN_IDs for each port are set, they display on the subcircuit schematic, and those names are used for the subcircuit pin names. The following example shows the PIN_IDs set.



4.7.1.2. Impedance and Hierarchy

In the Microwave Office program, when you use a schematic as a subcircuit, the PORT elements are only used for connectivity; the port impedance is NOT used. The port impedance is ONLY used when a simulation is performed on a top level schematic using ports. You can verify this by setting up a simple circuit with hierarchy and then varying the lower level's port impedance to see that it does not affect the top level schematic's response.

4.7.2. Using PORT_NAMEs

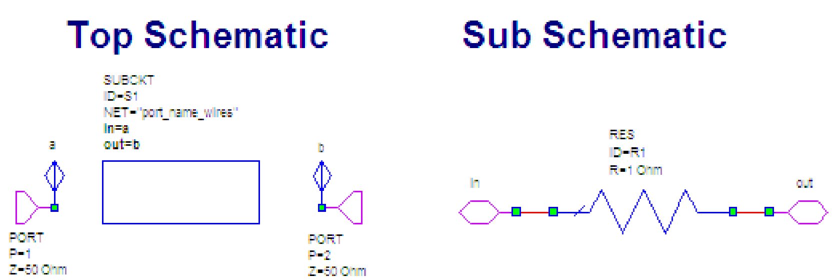
To add a PORT_NAME element to a schematic:

1. Click the **Ports** category in the Elements Browser.
2. Locate the PORT_NAME element in the lower window, click and drag it into the schematic window, position it, and click to place it.

You should use the PORT_NAME element when you build schematics that are used as subcircuits. These ports have only a Name parameter.

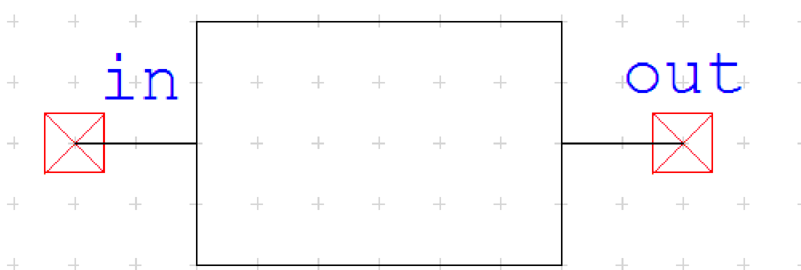
4.7.2.1. Hierarchy

When using a PORT_NAME schematic as a subcircuit, the subcircuit symbols behave differently than when not using the PORT_NAME element. The default subcircuit symbol has no nodes, and the connection names are available on the subcircuit symbols. The following example shows a subcircuit and a top level schematic.

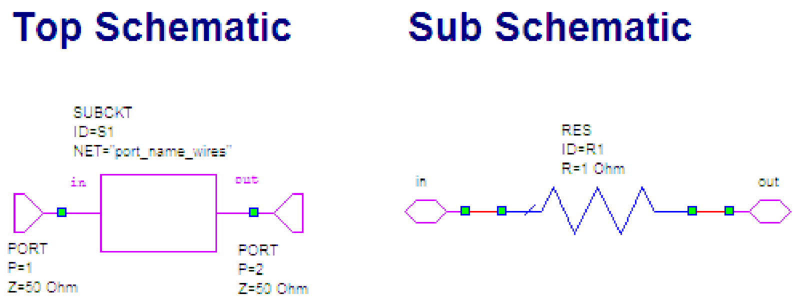


In the sub-schematic, there are two PORT_NAME elements, the left is named "in" and the right is named "out". When used as a subcircuit, notice they have no nodes to connect to. The names of the PORT_NAMEs display as parameters of the subcircuit, however. In this example, the connections to these ports are by named connectors. Named connections are discussed in more detail in "Connection by Name".

You can change the connections to be nodes on the symbol instead of by name, by creating a new symbol with node names that match the PORT_NAME names. For the previous figure, the following symbol shows the proper node names.



These names are case sensitive. If the subcircuit uses this symbol, the subcircuit and top level schematic display as follows.

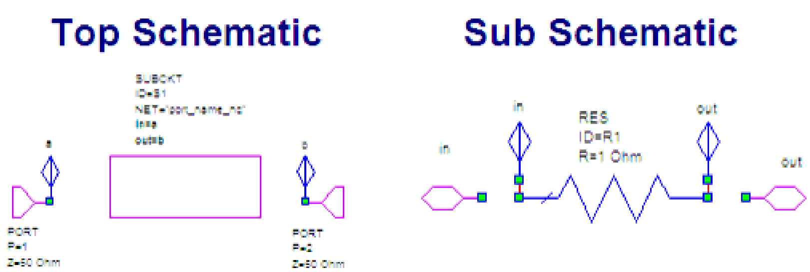


Note that the subcircuit in the top level schematic has nodes for connections, and the connection names as parameters of the subcircuits are no longer there.

When defining symbols, you define node names for only some of the PORT_NAME elements. For every match, you get a node. The other connections are by name.

4.7.2.2. Connection by Name

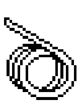
PORT_NAME elements can make connections with wires or connections by name. The previous figures show the PORT_NAMEs connecting using wires, however you can use the PORT_NAME Name parameter to make connections by name. For example, the previous figure is identical to the following figure where the PORT_NAME connections are made by name, not wires.



4.8. Connecting a Schematic or System Diagram

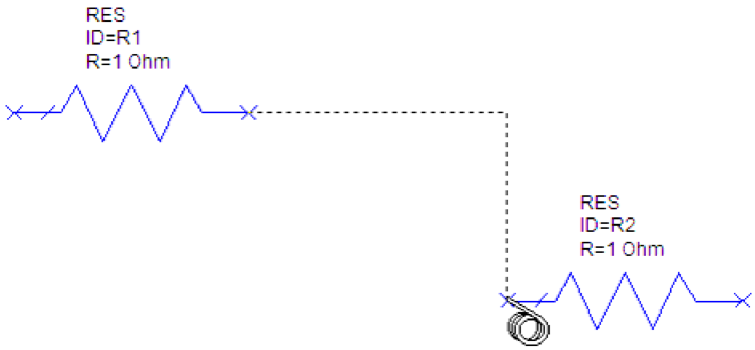
To design circuits and system diagrams, you connect elements and system blocks, respectively. You can connect elements by wires or by name, and connections can also be a bus (a single connection in a schematic that carries more than one signal).

4.8.1. Connection by Wires

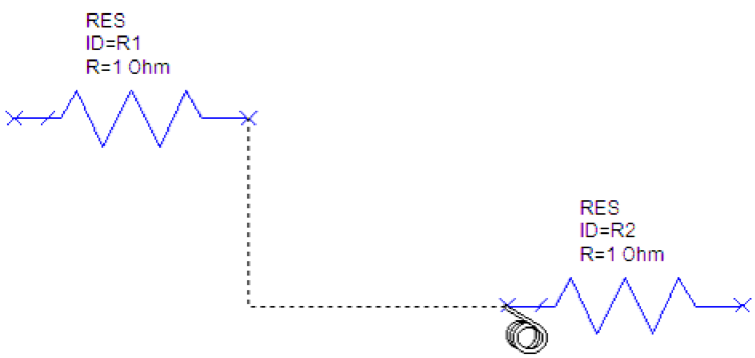


To connect two element or system block nodes with a wire, position the cursor over a node in the schematic or system diagram window. The cursor displays as a wire coil symbol.

Click at the point you want to start a wire and drag the mouse to the location where you want a bend, and click again. A dotted line displays showing where the wire will draw. For example, when wiring the resistors in the following figure, if you first move the mouse to the right, click, and then move the mouse down, the wire draws horizontally and then vertically.



If you first move the mouse down, click, and then move the mouse to the right, the wire draws vertically and then horizontally.



You can make multiple bends.

Right-click to undo the last bend point added.

Terminate the wire by clicking on another node or on top of another wire.

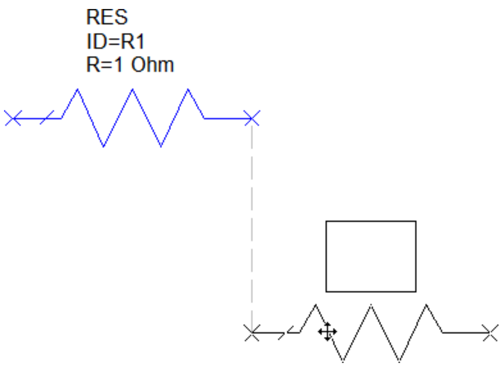
To cancel the wire, press the **Esc** key.

You can start a new wire from the middle of an existing wire by selecting the existing wire, right-clicking and choosing **Add Wire**, then clicking over the existing wire to start your new wire.

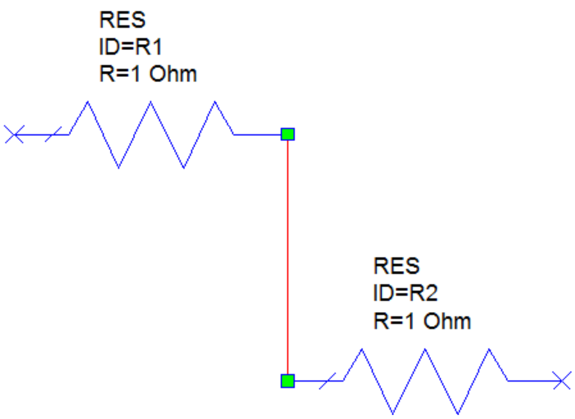
4.8.1.1. Inference Snapping and Auto-Wiring

Schematic/system diagram inference lines provide visual assistance in aligning elements and wires in the schematic/system diagram when adding new elements to a design, when pasting elements from the Clipboard, and when moving existing elements in the design. This feature can also aid in assembling schematic designs by automatically adding wires for aligned elements.

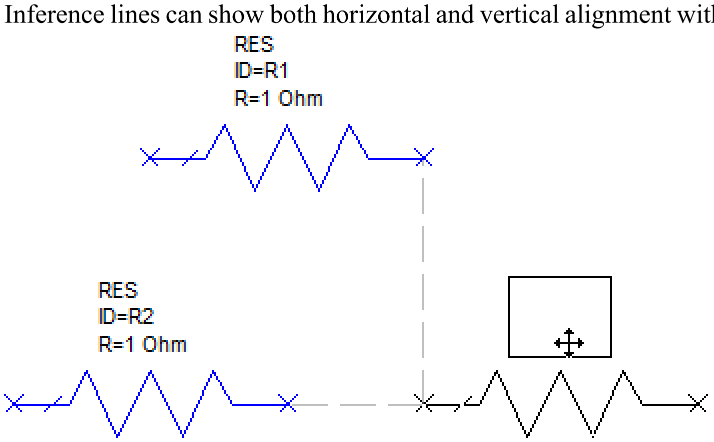
As shown in the following figure, when you add a new schematic element, when dragging the new element a faint gray dotted line displays when it becomes vertically or horizontally aligned with a node on an existing element.



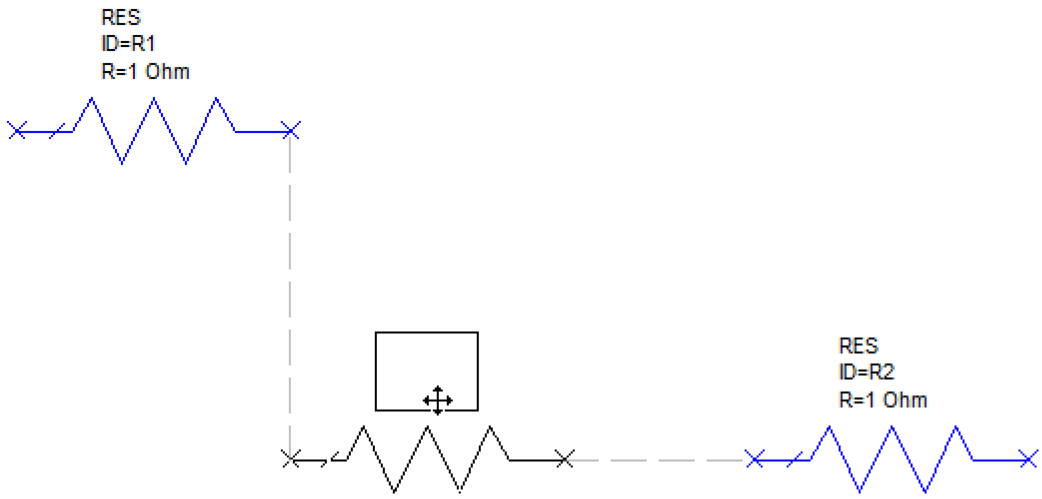
Adding the element at this point guarantees that you can connect the nodes using a straight wire segment. If you **Shift**-click to place the element, a wire segment connecting the elements along the inference line is added, as shown in the following figure.



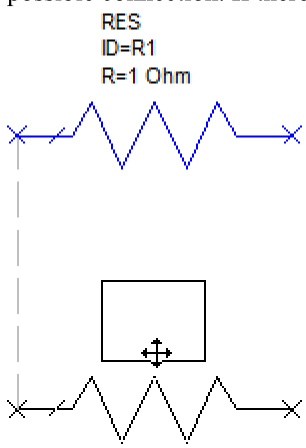
Inference lines can show both horizontal and vertical alignment with existing elements, as shown in the following figure.



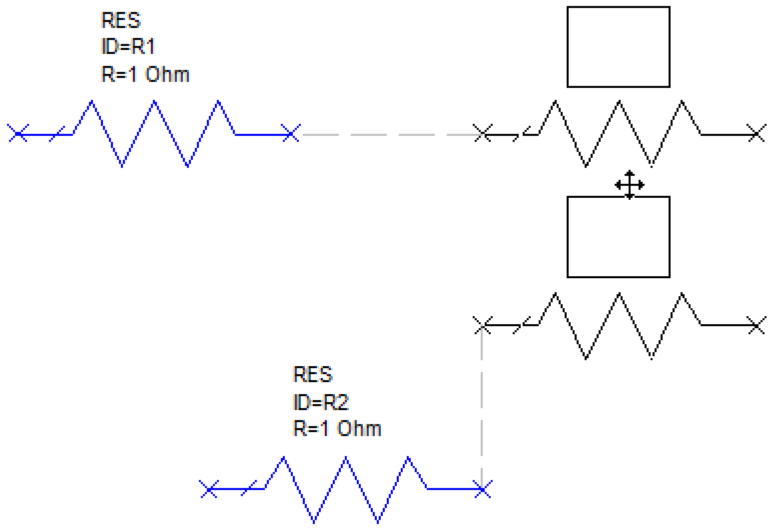
The lines can come from the same node or from different nodes on an element.



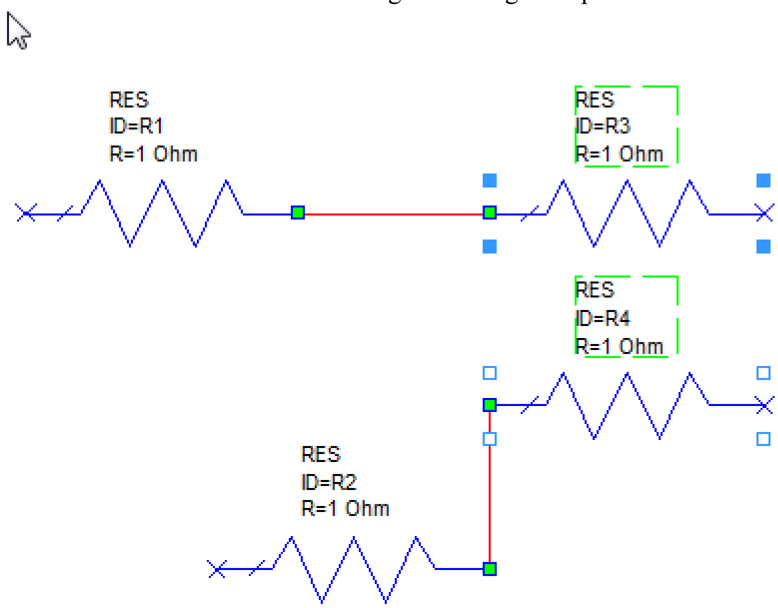
To avoid clutter and confusion, inference lines are limited to one horizontal and one vertical line drawn to the closest possible connection. If there are two equally likely connections, node 1 is typically preferred.



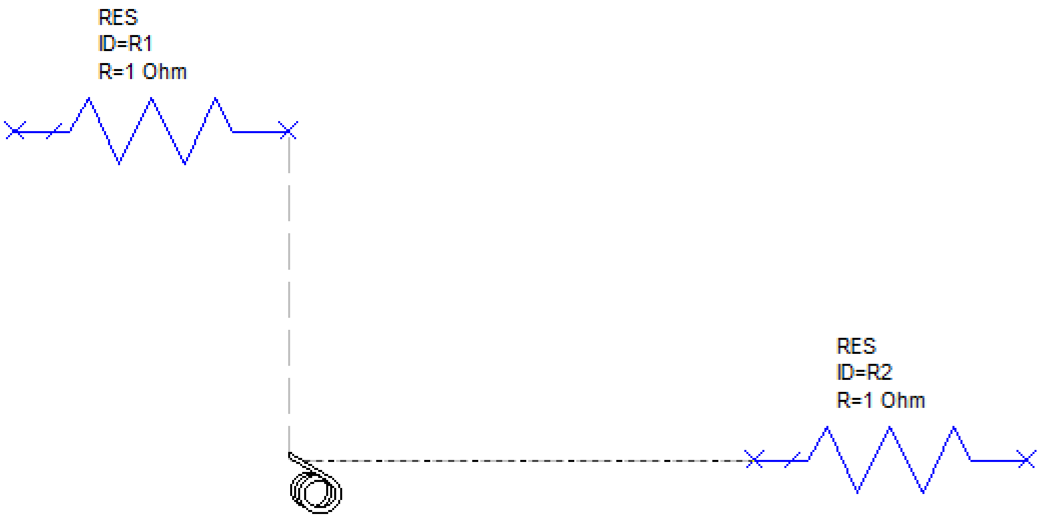
Inference lines can be drawn from multiple elements in a selected set to show the node alignments. In this case, the lines are drawn to the closest aligning nodes vertically, horizontally, or both.



You can also **Shift**-click when adding or moving multiple elements to connect the elements along the inference lines.



When you manually wire elements you can also **Shift**-click to complete wiring using the inference lines.



4.8.1.2. Connecting Many Elements or System Blocks

When you add wires, any wire that touches a node is connected to that node. You do not need to click on each node. For example, if you have 10 elements with their nodes in a line, you can add one wire that touches each element's node to connect all the elements.

4.8.1.3. Auto Wire Cleanup

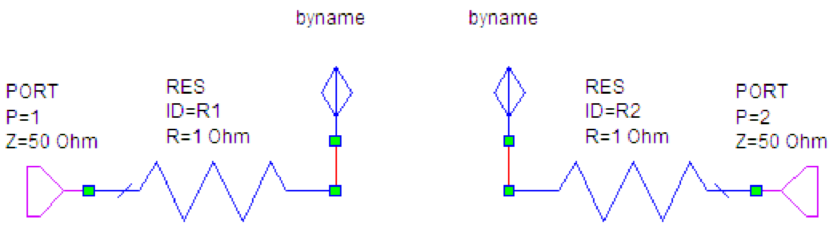
By default, when you delete an element or system block the wires connected to it are also removed. You can change this behavior and retain these wires. Choose **Options > Project Options**, click the **Schematics/Diagrams** tab, and clear the **Auto wire cleanup** check box to retain the connecting wires after you delete elements.

4.8.2. Element Connection by Name

You can also make element connections by name. This approach is typically used when adding wires to a schematic would make the schematic hard to manage. Both the PORT_NAME and NCONN elements allow you to specify connection names. The PORT_NAME element is discussed in "Using PORT_NAMEs".

The NCONN element is also used in schematics to make connections by name. This element is located in the Circuit Elements **Interconnects** category of the Element Browser. Its only parameter is Name. Any element nodes wired to NCONN elements or PORT_NAME elements with the same "name" are physically connected in the schematic.

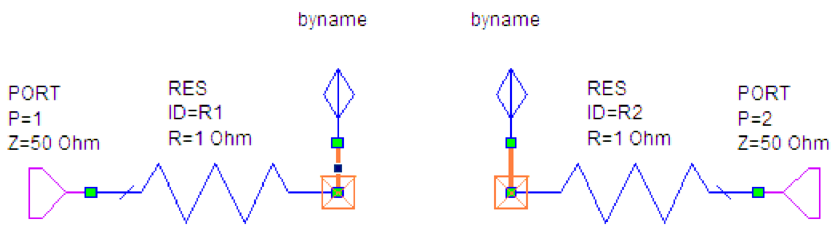
The following example shows two resistors connected by name using the NCONN element with the name "byname".



4.8.2.1. Verifying Connections

When elements are connected by name, you have no visual clues that the element's nodes are connected, as when using models. You can use the net highlight feature to help.

To use net highlighting, select any wire, right-click and choose **Net Highlight On**. Select a color from the dialog box that displays and then click **OK** to draw all wires and elements nodes that are connected with the specified color. The following figure shows net highlighting for the named connection between the resistors.



4.9. Copying and Pasting Schematics and System Diagrams

You can perform the following copy and paste operations on schematics and system diagrams and their nodes and subnodes:

- To copy and paste elements, system blocks, ports, and wires in the schematic or system diagram windows, select them and choose **Edit > Copy** and **Edit > Paste** or click the **Edit** and **Paste** buttons on the toolbar.
- To paste all or part of a schematic or system diagram into another instance of the same, click in the upper left corner of the area you want to include, then hold down the mouse button and drag the mouse toward the lower right until the area you want to include is enclosed in the box outline that displays, then release the mouse button. All elements or system blocks within the boxed area are selected. Choose **Edit > Copy**. With the target schematic or system diagram window active, choose **Edit >Paste**, move the copied objects to the desired location, and click to place them.
- To paste a schematic or system diagram into a Windows-based program as a metafile, with the schematic or system diagram window active, choose **Edit > All to Clipboard**. In the target application, choose **Edit > Paste**.
- To create a copy of a schematic or system diagram in the Project Browser, select the node you want to copy and then drag and drop it onto **Circuit Schematics** or **System Diagrams** as appropriate. A subnode with the rootname suffixed by "_1" is created for the first copy and incremented by one (_2, _3 and so on) for each additional copy.

You can right-click on a schematic or system diagram node in the Project Browser to access relevant commands such as those for renaming, exporting, deleting, or accessing schematic and system diagram options. You can also open another view of a schematic, document, or system diagram by choosing **Window > New Window** or by clicking the **New Window** button on the toolbar.

4.9.1. Adding Live Graphs, Schematics, Layouts, and System Diagrams

Schematics and system diagrams can contain other live schematics, system diagrams, or graphs; and in addition, schematics can contain layouts and 3D views.

To include one of these objects in a schematic or system diagram, simply drag the object from the Project Browser to an open schematic or system diagram window. When you release the mouse button a cross cursor displays. Click and drag the cursor diagonally to create a display frame for the added object. When adding another schematic, you can right-click and drag to display a menu with options for inserting it as a schematic, a layout, or a subcircuit. For more information about Window-in-window capabilities see "Window-in-Window ".

4.10. Adding Subcircuits to a Schematic or System Diagram

Subcircuits allow you to construct hierarchical circuits by including a circuit block within a schematic or system diagram. The circuit block can be a schematic, a netlist, an EM structure, or a data file.

4.10.1. Importing Data Files Describing Subcircuits

After adding a data file to a project, (see “Importing Data Files”), the data file displays under **Subcircuits** in the Elements Browser, and you can drag and drop it into a schematic or system diagram like any other subcircuit.

4.10.2. Adding Subcircuit Elements

A subcircuit can be one of many types including data files, netlists, schematics, and EM structures.

To add a subcircuit element to a schematic or system diagram:

1. Click **Subcircuits** in the Elements Browser. The available subcircuits display in the lower window pane. These include all of the schematics, netlists, system blocks, and EM structures associated with the project, as well as any imported data files added to the project.
2. To place the desired subcircuit, click and drag it into the schematic or system diagram, release the mouse button, position it, and click to place it.

Alternatively, you can add a subcircuit by clicking the **Subcircuit** button on the toolbar or by choosing **Draw > Add Subcircuit** or typing **Ctrl + K** from within a schematic to display the Add Subcircuit Element dialog box. For more information about this dialog box, see “Add Subcircuit Element Dialog Box”. Specify the required information, click **OK**, and then drop the subcircuit into the schematic or system diagram window.

For schematics as subcircuits only, you can right-click a schematic in the Project Browser and drag it into a schematic or system diagram window to add it as a subcircuit. Choose **Insert Subcircuit Here** on the menu that displays when you drop the schematic in the window.

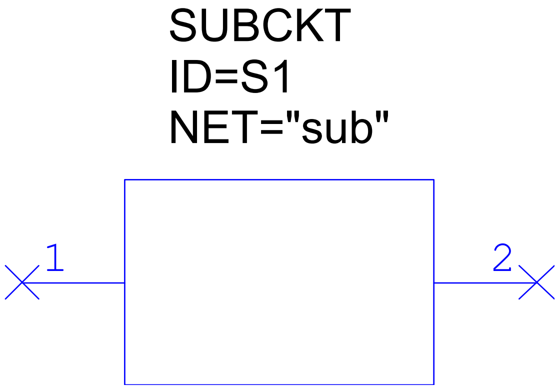
4.10.3. Subcircuit Grounding

When adding a subcircuit via the **Draw** menu or using the **Ctrl + K** hotkeys, you can decide how to handle the grounding type for the subcircuit. (For more information about this dialog box, see “Add Subcircuit Element Dialog Box”). After placing a subcircuit you can change its grounding type by double-clicking the subcircuit to display the Element Options: SUBCKT dialog box and clicking the **Ground** tab to view options. For more information about this dialog box, see “Element Options Dialog Box: Ground Tab”.

4.10.3.1. Normal Grounding Type

Normal is the default setting and the most common. In this case the ground used is ideal ground.

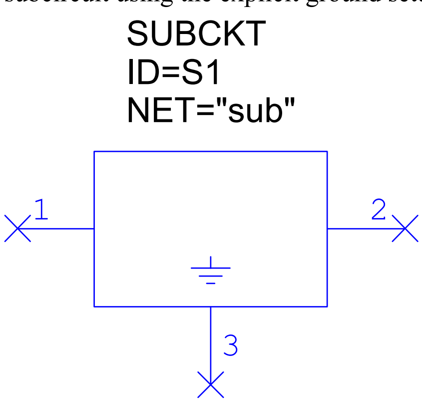
The default symbol in this mode is a box with the nodes spaced around the outside of it. The following figure shows a two-port subcircuit.



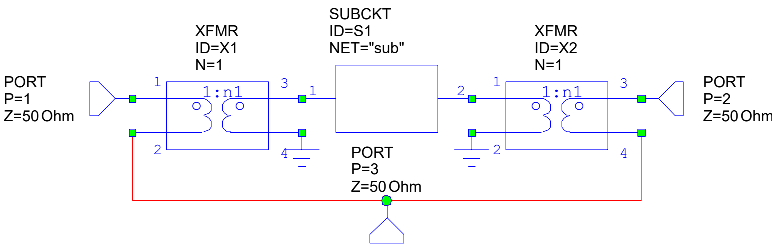
4.10.3.2. Explicit Ground Node Grounding Type

An explicit ground node can be viewed as a local ground for the subcircuit file. It is important to understand that the same ground is used for all ports in the subcircuit. This implies that physically, the structure is electrically small, or has a very good (perfect) internal grounding system connecting the ports. Normally, exposing the ground node is used for transistor data, where a common ground node in the measurement is being exposed.

The default symbol in this mode is a box with the nodes spaced around the outside of the box and one additional node with a ground symbol to indicate which node is exposing the ground node. The following figure shows a two-port subcircuit using the explicit ground setting.



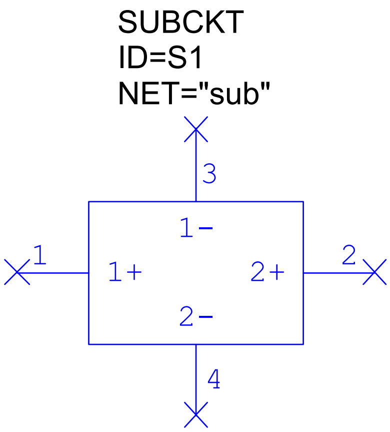
To better understand the explicit ground node, the following diagram shows how you can recreate the explicit ground setting using transformers.



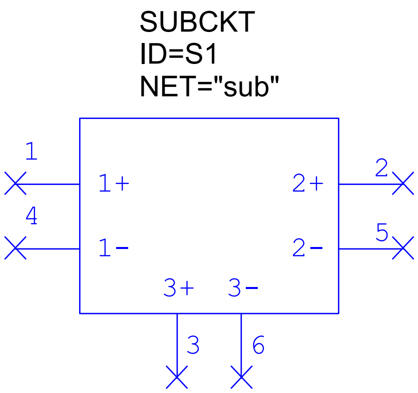
4.10.3.3. Balanced Ports Grounding Type

Balanced ports extend the exposed ground node concept by creating a local ground node for each port. It is possible to misuse this concept so you should familiarize yourself with the following sections on limitations.

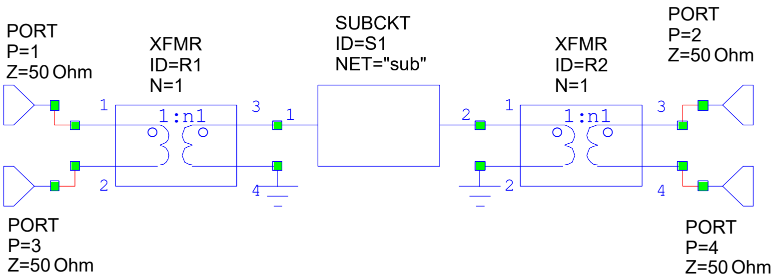
The default symbol in this mode is a box with the nodes spaced around the outside of the box. The symbol has text indicating which node is the positive and negative terminal of the port. The following figure shows a two-port subcircuit using the balanced ports setting.



For more than two ports, the terminals for each port are along the same side of the symbol. The following figure shows a three-port subcircuit using the balanced ports setting.



To better understand the balanced ports, the following diagram shows how you can recreate the balanced setting using transformers.



4.10.3.4. Proper and Improper Ground Usage

Exposing the ground is a perfectly valid concept when properly used, but, unfortunately, it is often misapplied. As an example, assume you start with a two-port S-parameter file. By exposing the ground, a three-port S-parameter file is obtained, with the third port being the "local ground" return. Remember, the assumption is that the local grounds of the ports are electrically the same (they are connected to each other by a very, very good ground return). The exposed node connects to that ground return. By doing this, for example, an engineer can DC bias the ground return. Typically, this procedure is used for transistor S-parameters. Transistors have three ports, but when measuring a transistor with a network analyzer, one port is grounded, and a two-port S-parameter file results. By exposing the ground node of the S-parameter file, a designer can attach elements to the previously grounded port, for example, an inductance to the common source node. This concept can also be used where the transistor is housed in a package and the global circuit ground is attached to the "transistor package's ground." The key point is that this method works because the local grounds of the ports are essentially the same, and are attached to each other by a perfect grounding structure.

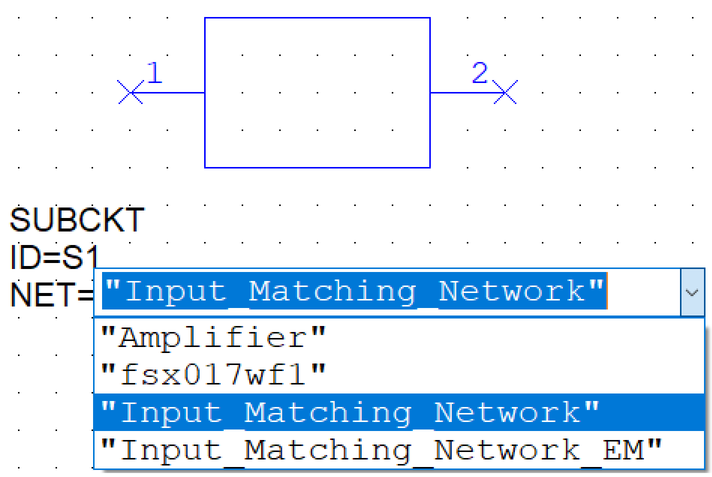
A common mistake is to assume that imperfect ground properties can be observed by looking at the exposed node, for example, the loss of the ground plane. The exposed ground approximation assumes the ports' local grounds are the same; however, imperfect ground properties would make the ports' local grounds different voltages. There is even greater confusion with multi-port S-parameter files, where differential ports, or local grounds, are requested. For example, a two-port S-parameter file will now have four ports, with Port 3 corresponding to Port 1's "ground," and Port 4 corresponding to Port 2's "ground." This can be a useful tool, but unfortunately is misunderstood by many designers who incorrectly assume they are looking at the "local ground" of the port. For example, they mistakenly think they can measure the resistance of the original lossy ground by placing an Ohmmeter across the two new "ground" ports. The original S-parameter file did not have this information (it assumed the local grounds were at the same voltage!). These ports were created by the mathematical operation of adding transformers. A math trick cannot recover lost physics, no matter how hard a designer tries.

4.10.4. Editing Subcircuit Parameter Values

To edit subcircuit parameters:

1. Click on the subcircuit in the schematic or system diagram window to select it.
2. Right-click, and choose **Edit Subcircuit**. The subcircuit opens in the workspace.
3. Double-click an entity in the subcircuit to display the Element Options dialog box. You can edit subcircuit entities like any other entities, as described in “Editing Element Parameter Values”.
4. To change the grounding type of the subcircuit, click the **Ground** tab in the Element Options dialog box and select the appropriate type. Ground type may only be specified for one or two port subcircuits.

You can edit the name of the network referenced by the subcircuit directly on the schematic. To edit the NET parameter on the schematic, double-click this parameter name and then select the name of the network to use.

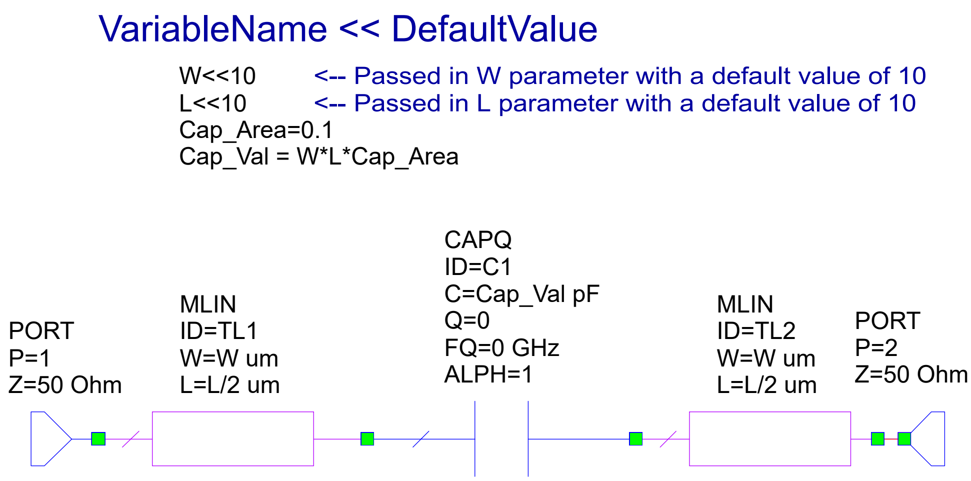


4.10.5. Using Parameterized Subcircuits

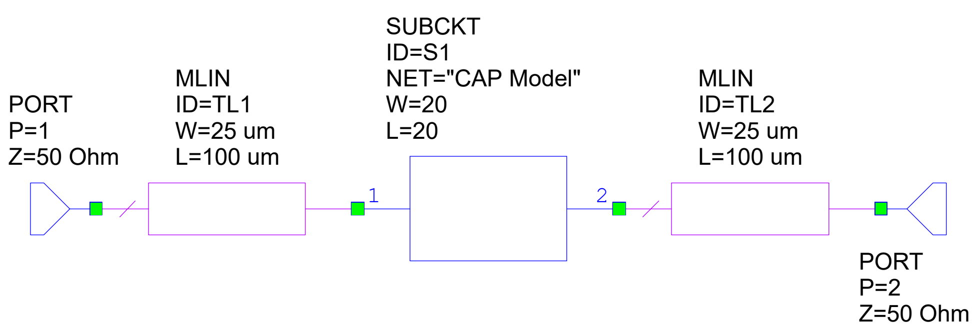
The AWR Design Environment platform supports parameterized subcircuits, which allow subcircuits to use values passed in from system diagrams or schematics at a higher level in the hierarchy. Creating a parameterized subcircuit requires creating variables and equations to express the parameter values. The following Microwave Office example shows the types of variables and equations that you must create. For details on creating variables and equations, see "*Variables and Equations*".

This Microwave Office example demonstrates how to create and use a parameterized subcircuit. The example shows a model of a simple thin-film capacitor implemented as a parameterized network. A passed-in parameterized variable is created using the following syntax:

```
VariableName << DefaultValue
```



The following figure shows the use of the parameterized subcircuit within another schematic. You can change the values passed in by editing the W or L parameter of the subcircuit. The passed-in variables do not have units, but when they are assigned to an element that has units, they use the element units. This is consistent with variable use within the schematics. For example, if the passed-in value for W is 20, then the value used for the W parameter of TLI (in the previous diagram) is 20 um.

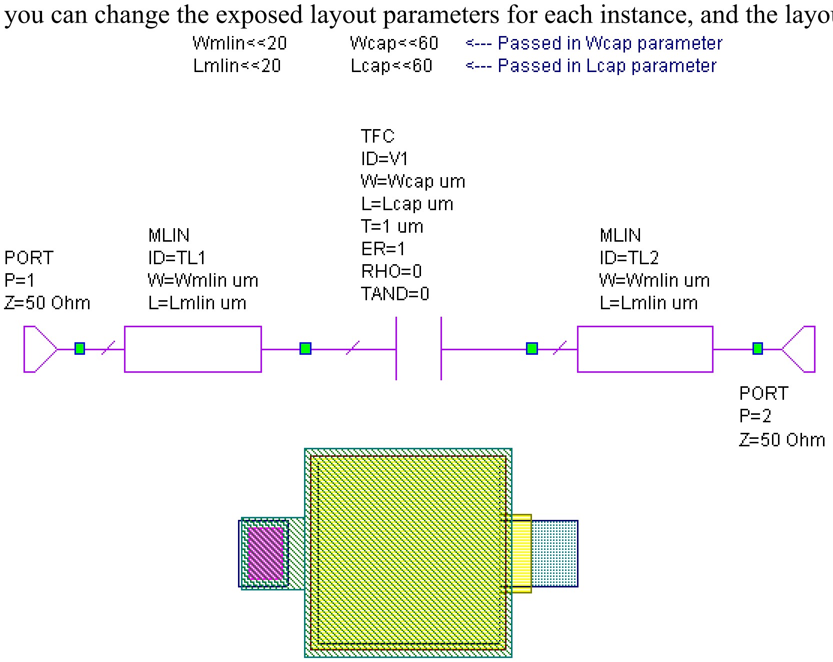


When parameterized subcircuits have passed-in variables that *do* have units set (in the Edit Equation dialog box), equations **Unit Type** is not set to **Scalar**, and dependent parameters *do not* use base units, you can select **Scale exported parameters with units to user unit values** on the Options dialog box Schematic tab to scale the values passed in from the parent to user units (project units as opposed to base units) in the child. Starting in V13, passed parameters with units are scaled correctly with this option set.

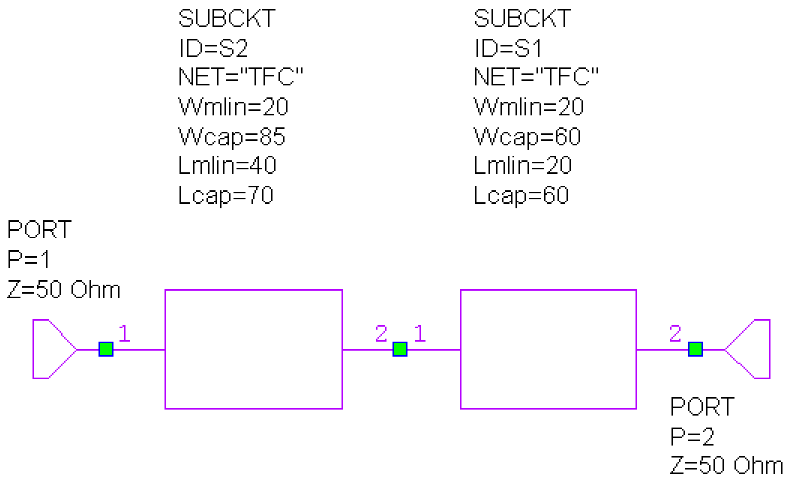
By default, **Scale exported parameters with units to user unit values** is selected, but for projects prior to V13, the software checks to see if there are any documents that are not set to use base units, and that have passed parameters with units. If this combination of events is detected, this option is cleared for every document where this scenario is detected to prevent changing the behavior of any documents that may have applied manual rescaling of the equations in previous versions of the software. If you want to enable this option for a document in which it is auto-disabled, you may need to rewrite some of your equations to take care of the difference in unit scaling.

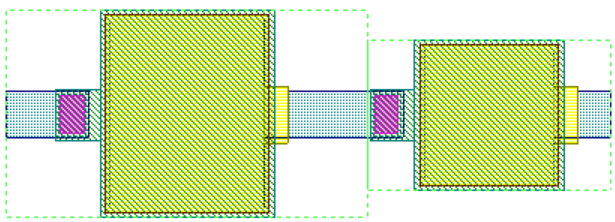
4.10.5.1. Using Parameterized Subcircuits with Layout

Parameterized subcircuits draw the geometry by scaling the subcircuit parameters that affect the layout. By default, a parameterized subcircuit has the same layout as the subcircuit. To create a parameterized subcircuit with layout, simply create a passed-in variable for parameters that affect layout. For example, the Thin Film Capacitor with layout cells in the following figure has parameters L and W as passed parameters. When this subcircuit is used in another schematic, you can change the exposed layout parameters for each instance, and the layout changes accordingly.



The subcircuit shown in the previous figure is used twice in another schematic, as shown in the following figure, and its passed parameters are altered. The layout updates accordingly.





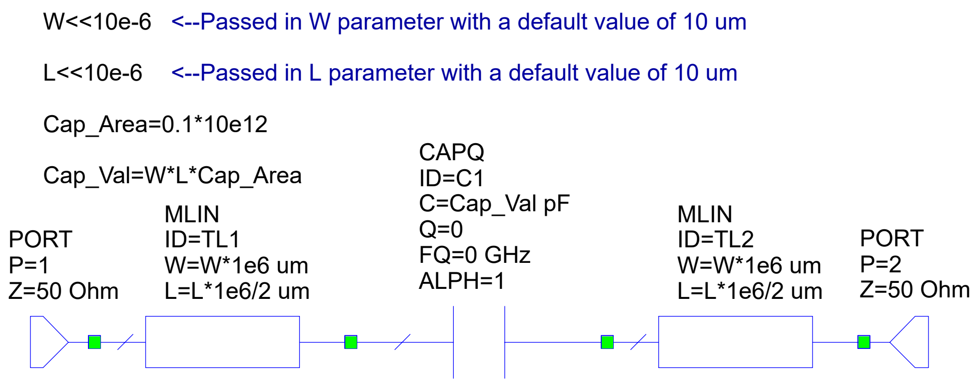
If the parameters passed into the subcircuit do not affect layout, the subcircuit's default layout displays. For example, capacitor dielectric thickness or process corners.

To prevent a parameter from affecting the layout:

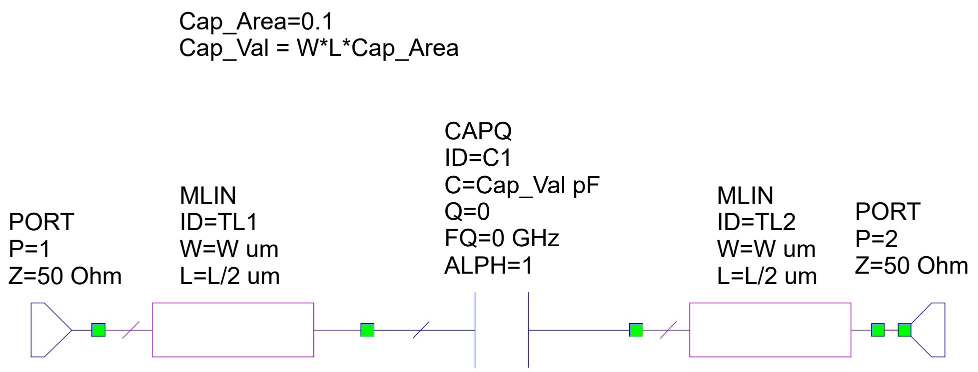
1. Right-click the parameter equation and choose **Properties** to display the Edit Equation dialog box. See "Edit Equation Dialog Box" for more information.
2. Select **Parameter definition** under **Variable Type**.
3. Select the **Does not affect layout** check box under **Parameter Description**.

If the element does not have an associated layout, you can also use the Cell Stretcher to create the parameterized subcircuit with layout.

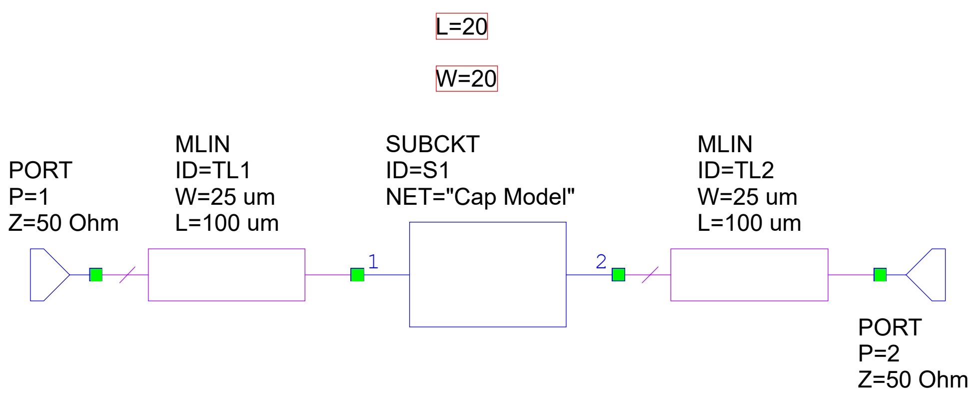
For a parameterized layout cell, the passed-in parameters are unitless, so the parameterized subcircuit should be designed to have unitless parameters passed in, to make it compatible with existing layout cells. For example, the following parameterized subcircuit is designed to work with um length units (1e-6 m).



As shown in the following figure, you need to specify the values of W and L in meters for the subcircuit instance.



Select the subcircuit, right-click, choose **Properties** and then click the **Layout** tab to assign the subcircuit instance a layout cell to draw that uses the W and L parameters. If you choose the TFC (thin-film capacitor) layout cell, the following layout is produced from the previous circuit (read in the *MMIC.lpf* file in the program directory */Examples*subdirectory to re-create this example).



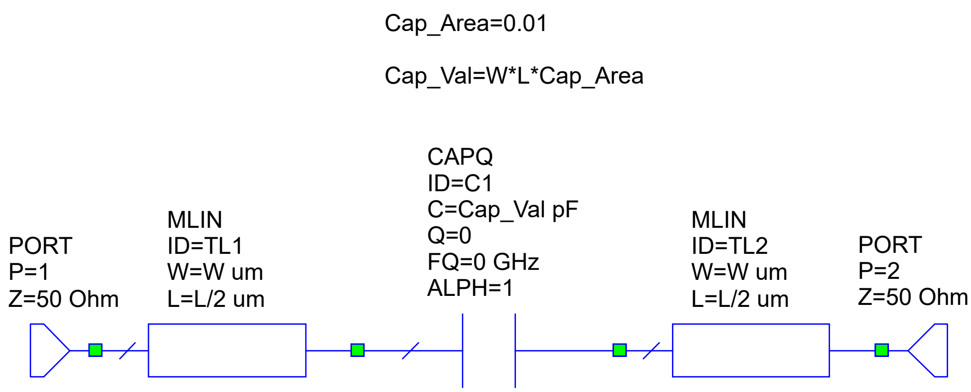
You can create parameterized layout cells using the Layout Cell Wizard. For more information on layout cells, see *"Layout Overview "*.

For a Cell Stretcher, you set up the artwork cell and Cell Stretcher objects as described in "Stretching Artwork Cells", and then assign the subcircuit's layout as the newly created artwork cell configured for cell stretching.

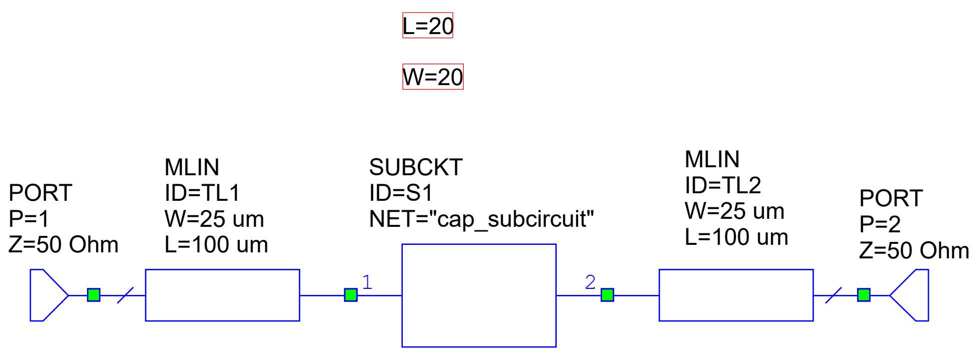
4.10.6. Using Inherited Parameters

An alternative to using the parameterized equation syntax for specifying parameterized subcircuits is to set variables as inherited parameters. An inherited parameter is a parameter that is passed down through hierarchy from the top level without requiring you to explicitly pass the parameter from each subcircuit. To change a variable inheritance value you can use the Tune tool and press **Ctrl** (weak) or **Shift** (strong) when you select the variables. Alternatively, you can change the **Pass Down Mode** in the Edit Equation dialog box (see "Edit Equation Dialog Box"). The setting is indicated by a red box around the variable. A solid red box indicates strong dependency and a dashed box, a weak dependency. The following example demonstrates how the values for W and L display when strong and weak dependencies are set.

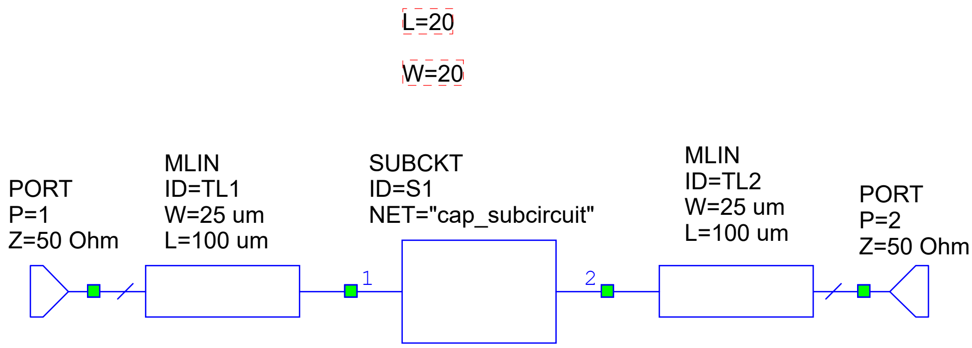
The following figure shows the subcircuit where the variables "L" and "W" are used, but not defined on the schematic.



The following figure shows the "L" and "W" parameters being passed down strong to the subcircuit.



The following figure shows the "L" and "W" parameters being passed down weak to the subcircuit.



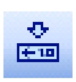
Note that subcircuit variables with a strong dependency override any other variable values from that level and down the hierarchy. Subcircuit variables with a weak dependency are only used if there is no other definition of that variable.

Variables defined in the Global Definitions window do not have an inheritance setting; they effectively always have weak inheritance.

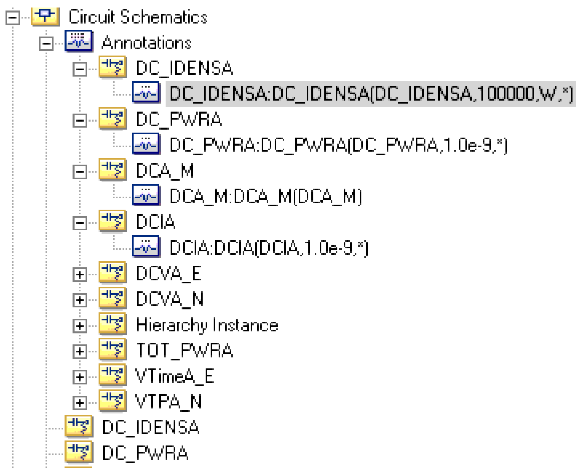
4.11. Adding Back Annotation to a Schematic or System Diagram

Back annotation displays DC current, voltage, total power, current density and other measurements directly on a schematic, and VSS signal and node, diagnostic, and fixed point measurements on a system diagram. For example, if you are performing a nonlinear simulation, you can back-annotate the DC voltages at each node in the circuit to ensure your circuit is biasing properly. You can also annotate results from Harmonic Balance or Cadence APLAC® HB simulation.

To access back annotation, right-click on a schematic or system diagram node in the Project Browser and choose **Add Annotation** to display the Add Annotation dialog box to select a measurement and specify settings. See "Add/Edit Schematic/System Diagram/EM Structure Annotation Dialog Box ". You can also click the **Annotation** button on the toolbar.

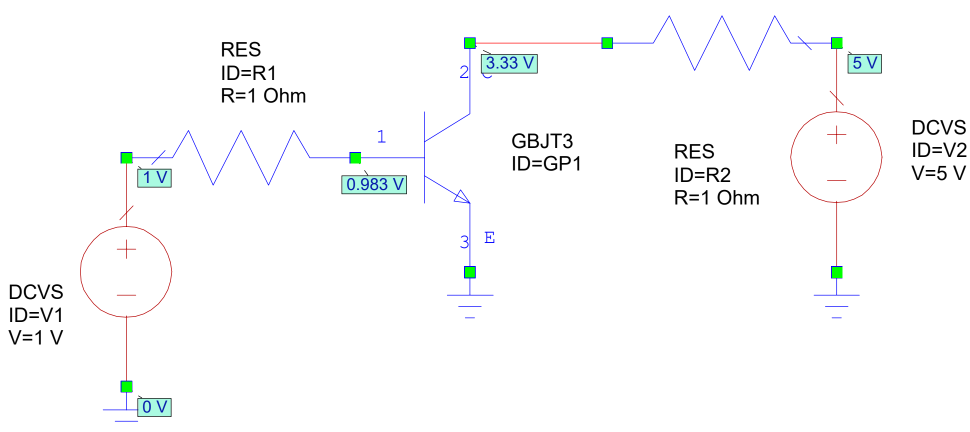


In an **Annotations** node beneath the **Circuit Schematics** or **System Diagrams** node, each annotated schematic is listed with its annotation below it. You can double-click an annotation to modify its settings. You can add multiple annotation types to a single schematic or system diagram.



After adding annotations, you can right-click on the schematic or system diagram node and choose from options for enabling/disabling/toggling all annotations.

The following figure shows a schematic with back annotation.



4.12. Vector Instances, Buses, and Multiplicity

Vector instances, buses, and multiplicity are advanced schematic techniques to help keep a schematic manageable when creating large designs.

4.12.1. Vector Instances

A vector instance is a means of having one schematic element that defines several instances of the element for simulation and layout.

You define an element's vector instance by typing the vector instance notation at the end of each element's ID parameter directly on the schematic. The syntax for vector instances is "[0:N]", where N is the N+1 number of vector instances. For example, [0:3] defines four vector instances of that element.

When you use vector instances, it is identical to creating N number of models on the schematic, increasing the number of elements that get simulated.

4.12.2. Buses

A bus is a schematic connectivity feature that contains more than one wire (similar to a vector instance of a schematic wire). Buses are commonly used to connect elements using vector instances.

Buses are automatically created when you wire to elements using a vector instance. You can also select any wire, right-click and choose **Create BusNet**.

With a bus, the wire has a name and the syntax "[0:N]" where N is the N+1 number of wires in the bus. You can edit the bus name to change N or remove the text in braces to remove the bus.

When you change the vector instances of an element, the number of wires in the buses attached to the element are normally automatically updated to match the vector instances. However, once you edit the name of a bus the bus is no longer automatically updated. You need to edit the bus to update the number of wires.

The bus is identified by a name, so you can make connections by name. For this reason a bus does not need to be connected in the schematic.

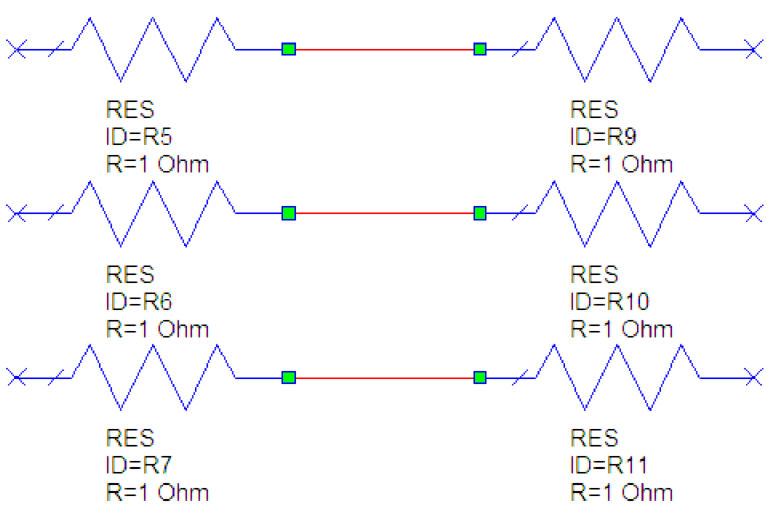
Individual connections to a bus are made by named connectors or PORT_NAME elements. The name used defines which bit of the bus to use. If you have a three-bit bus, for example B[0:2], a named connector with the name B[0] connects to the first bit of the bus, B[1] to the second bit, and so on.

4.12.3. Connectivity with Vector Instances and Buses

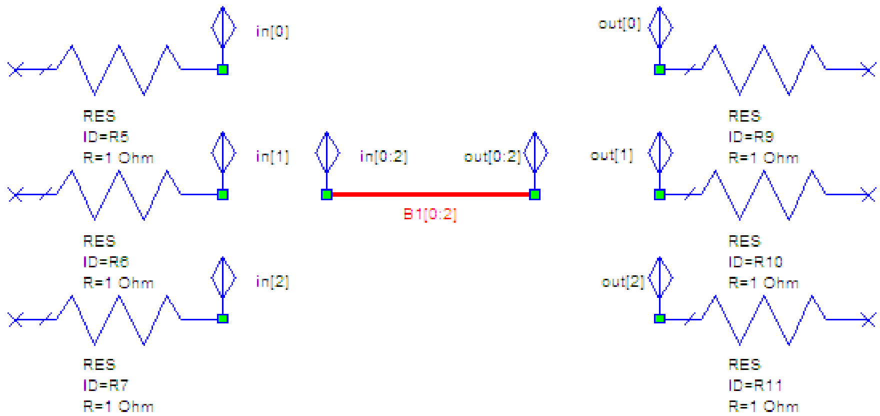
There are several ways you can use vector instances and buses together.

4.12.3.1. Separated Elements and Wires

The following example uses two sets of connected resistors as a starting point. All of the examples have the same connectivity, although it is implemented differently. The following figure shows a schematic using traditional wires.

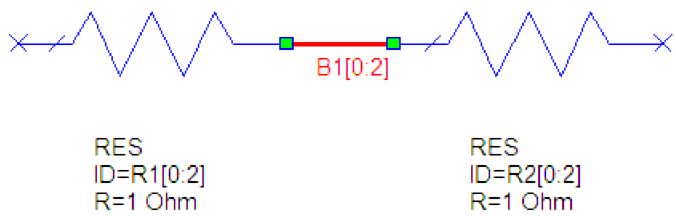


The connectivity between the resistors can be via a bus, as shown in the following figure.



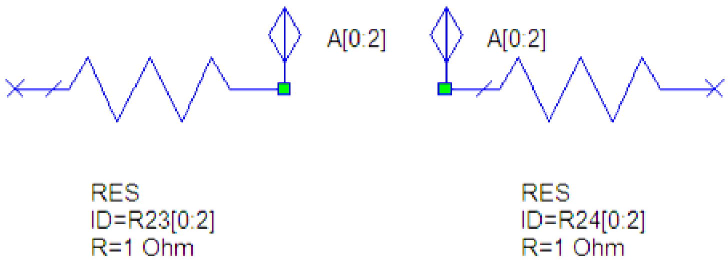
Here there is a bus named "in", "out" and the two buses are connected by the bus "B1".

You can connect the resistors themselves using vector instances as shown in the following figure.

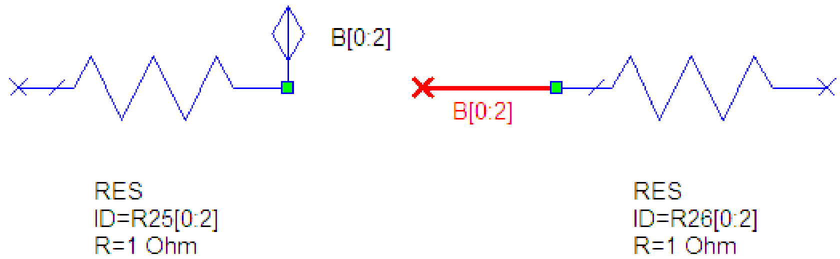


Note that the bus is connected to elements with vector instances and the bus is directly wired to the node of the element.

You can also make this connection with named connectors as shown in the following figure.



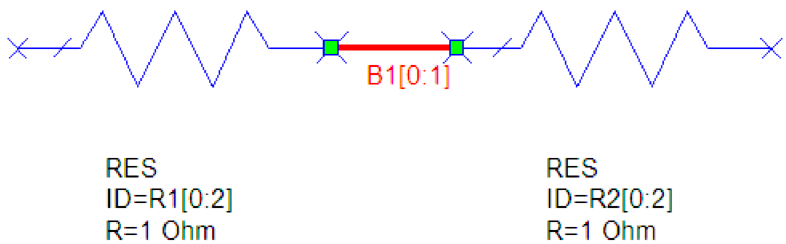
The bus itself has a name that you can use for connectivity, as shown in the following figure.



4.12.3.2. Bus and Vector Instance Sizes

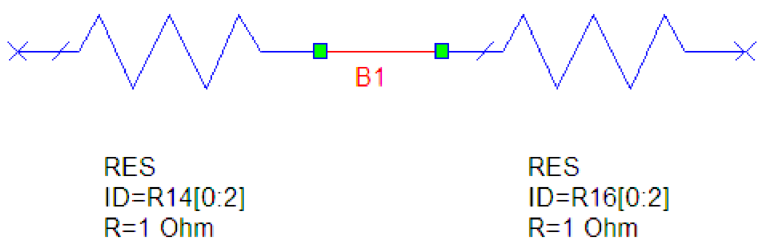
When wiring vector instances, the connection can only be a single wire or a bus the same size as the vector instances.

If a bus is a different size than the elements connecting to it, the bus shows that it is not connected to the element. The following example is the resistor example above, except the bus size is now 2 instead of 3.

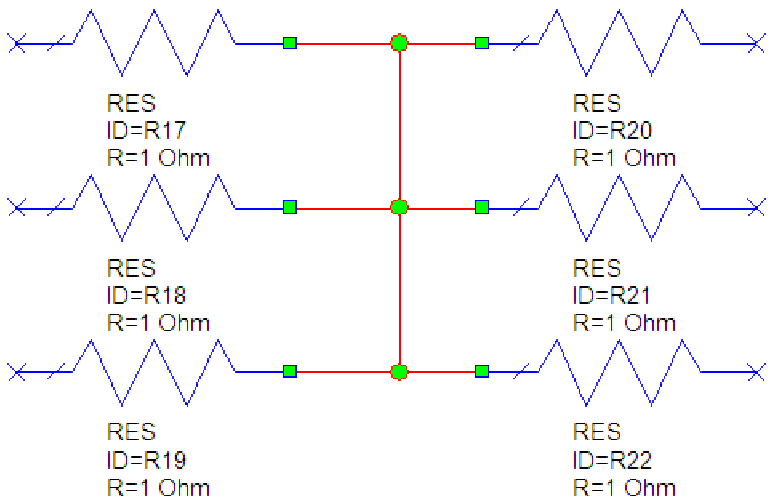


Notice that the element nodes where the bus is connecting display as X's. If the connection is correct, these display as squares.

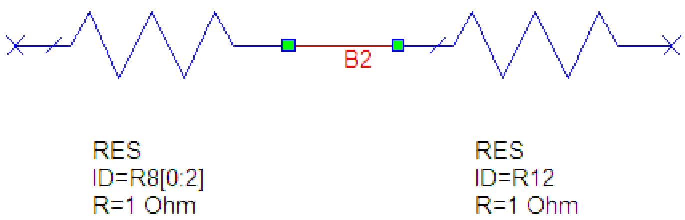
All of the previous examples showed a bus the same size as the vector instances on either size. If you use a wire instead of a bus, all of the vector instance elements are shorted together. The following figure shows the same example with vector instances using a wire.



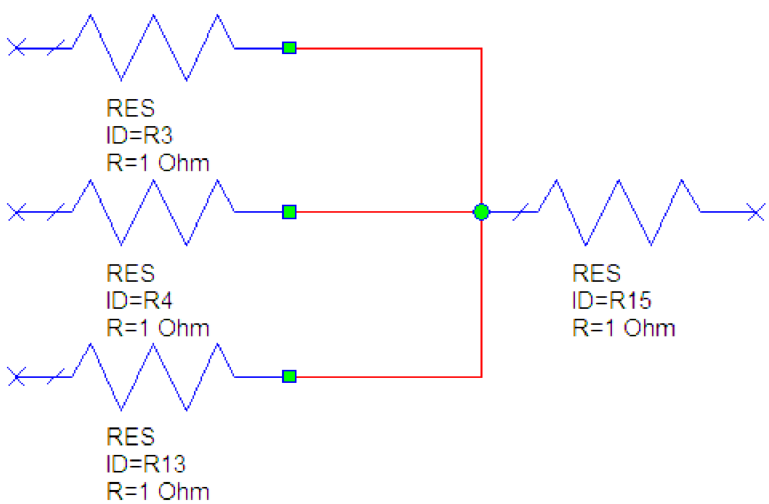
This is equivalent to the following figure.



If you connect an element with vector instances to an element without vector instances, you should always use a wire instead of a bus. The following figure demonstrates this scenario.



This is equivalent to the following figure.

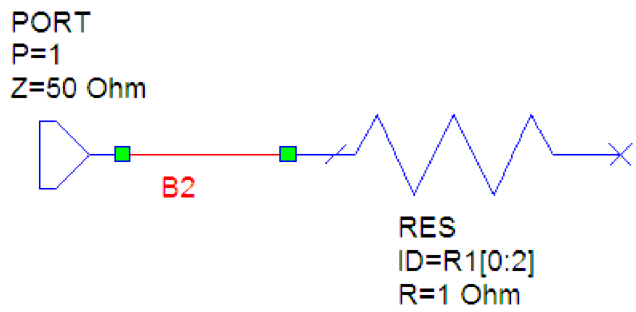


You might commonly use this when grounding a vector instance, as only one ground element is required.

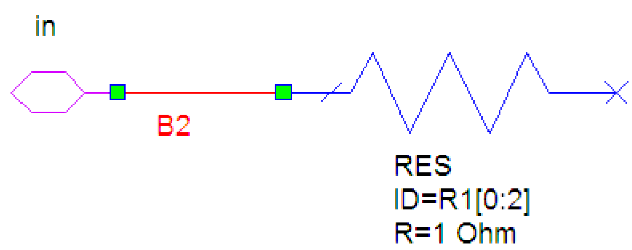
4.12.3.3. Using Ports

When connecting elements with vector instances to ports there are several options.

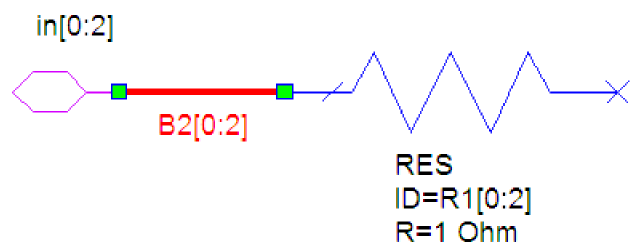
If you are using any port except PORT_NAME, the vector instance is shorted on the port so you must use a wire to connect to the port.



If you are using the PORT_NAME port and are not using vector instance notation, you should also use a wire to connect and short the vector instance at the port.



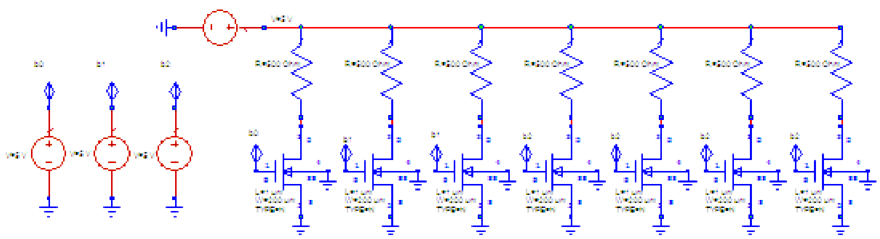
If you do not want the vector instances to be shorted at the port, use vector instance notation for the name for the PORT_NAME to maintain the bus connection through hierarchy.



When using a PORT_NAME through hierarchy, if you want the subcircuit to have a node to connect to, the symbol node name must match the name of the PORT_NAME. This means the symbol node name requires the vector instance notation.

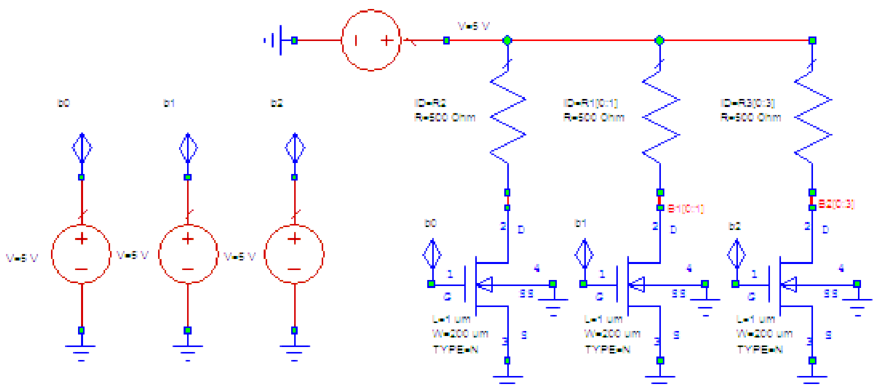
4.12.3.4. Bundles

A bundle is a collection of signals to use on a bus. For some buses, all the signals come from a vector instance of the same size. For others, the signals might come from different locations in the circuit. To demonstrate, the following shows a simple 3-bit control used to control the total current in a circuit. Identical sized transistors are used. There are three separate voltages named b0, b1, and b2 controlling the transistors. The voltage b0 controls one transistor, b1 controls two transistors, and b2 controls four transistors. To create this schematic without using vector instances or buses, see the following figure.



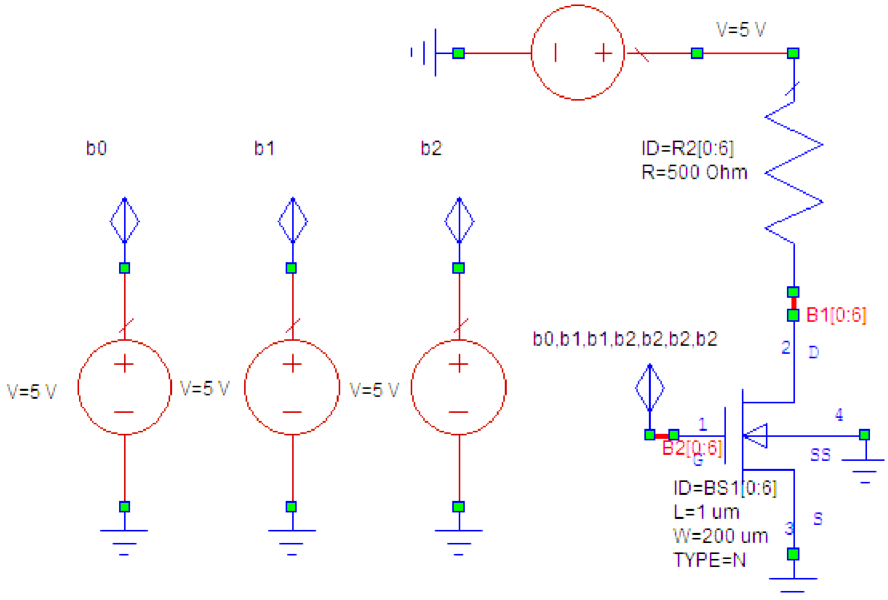
Note that the connection to the input of each transistor is described for each controlling voltage.

The following is a simpler schematic created using vector instances for the transistor and resistor for each controlling bit.



Notice that two of the transistor and resistor pairs are using vector instances and buses. The named connector at the input, however, is a single wire shorting each input to the control voltage.

This schematic can be further simplified as follows.



Now there is only one transistor and resistor both using vector instances and buses to produce the seven pairs. The input to the transistor is now using a 7-bit bus. The connections to the bus are defined by a comma-separated listing of named connections, b0,b1,b1,b2,b2,b2,b2, which is called a bundle.

The bundle notation in this example can be further simplified. Typing **b0,2*b1,4*b2** accomplishes the same connectivity.

4.12.3.5. Buses in VSS

Because wires between VSS blocks represent signal paths and have direction, VSS software provides a number of blocks to help convert between a single signal and a bus:

BUS_SPLITTER	Similar to the RF Splitter block SPLITTER except the split output signals are in a bus; this block may also be reversed and used as a combiner.
BUS2MUX	Similar to the Multiplexer block MUX except the signals to be multiplexed together come from a bus.
BUS2SER	Similar to the Parallel-to-Serial Converter block P2S except the parallel signals come from a bus.
MUX2BUS	Similar to the De-Multiplexer block DEMUX except the de-multiplexed output signals are in a bus.
PHARRAY_RXSIG_BUS	Similar to the Phased Array Signal Splitter for Receiver block PHARRAY_RXSIG except the output signals to the antenna elements are in a bus.
SER2BUS	Similar to the Serial-to-Parallel Converter block S2P except the parallel signals are in a bus.
SIG2BUS	This block copies the input signal to all the output signals of the bus.

NOTE: If you are trying to replicate an input signal onto a bus as-is, the preferred block is SIG2BUS and not BUS_SPLITTER. BUS_SPLITTER is an RF block, which includes additional overhead for supporting the RF capabilities.

4.12.4. Multiplicity

Multiplicity is a means for an individual element or a subcircuit to scale the response of the element by the multiplicity factor *M*. Some nonlinear models such as BSIM have a multiplicity parameter built into the model.

Any time you use a subcircuit, it has a secondary multiplicity parameter. When you use multiplicity, the equations that are solved are scaled by the multiplicity factor.

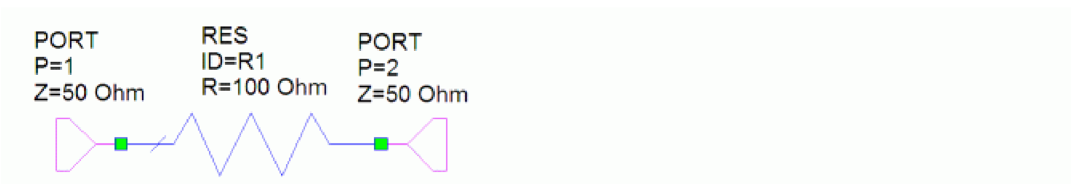
For layout, on the Element Options dialog box **Layout** tab you can specify what model parameter determines multiplicity. When properly configured, multiplicity creates M copies of the element layout.

4.12.4.1. Vector Instances Versus Multiplicity

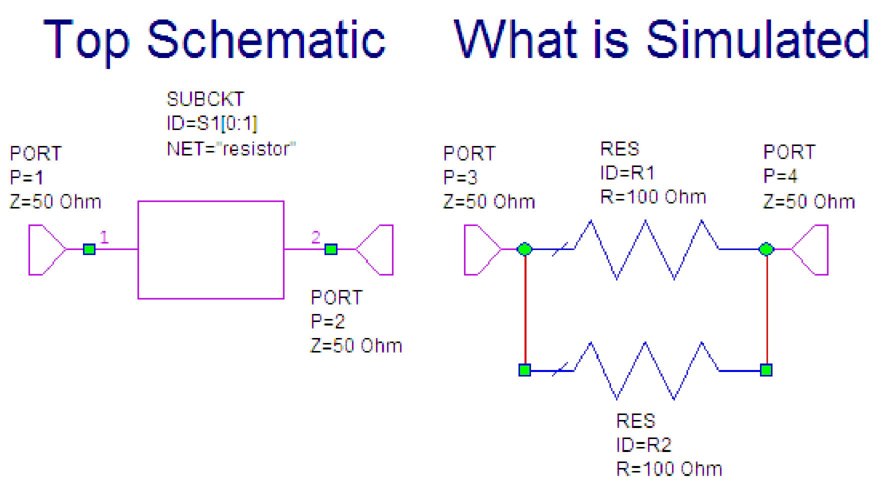
Vector instances treat each instance of the element as a separate model. Simulation time and memory use grow when adding vector instances.

Multiplicity scales the problem to be solved. Simulation time and memory use do not grow when using multiplicity, compared to elements without it.

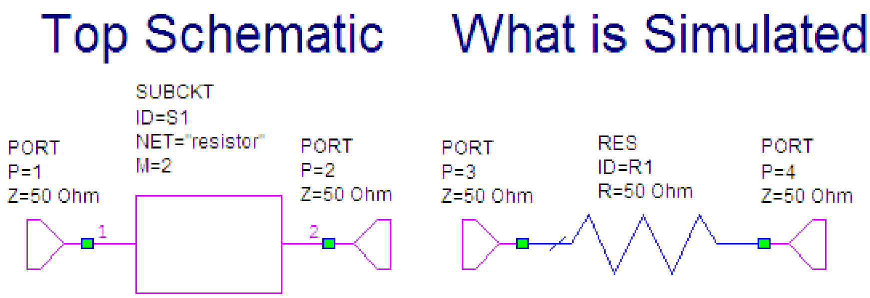
To demonstrate, the following uses a resistor as a subcircuit that uses vector instances and multiplicity to compare the differences. The subcircuit is a 100 ohm resistor.



The following figure shows the resistor used as a subcircuit using vector instances, and a view of what the simulators must solve.



The following figure shows this resistor used as a subcircuit using multiplicity, and a view of what the simulators must solve. Notice the resistance value is different.



Using Vector Instances or Multiplicity

Based upon the desired outcome, you must decide whether to use vector instances or multiplicity. The most important concept to remember is that vector instances increase the simulation time while multiplicity does not.

Multiplicity doesn't work with buses the way vector instances do. You can never use a bus to connect two subcircuits using multiplicity. The nodes of any element using multiplicity are always connected.

You should consider this carefully when using iNets. For multiplicity, you can get multiple copies of the layout cells. You can route an iNet to each one of these cells. When simulated, the ends of the iNets connecting to the element layouts are all shorted. You may get total net capacitance correct this way, but resistance and inductance will not be correct. Vector instances allow iNets to be wired to each layout instance and properly model the connection of the iNet.

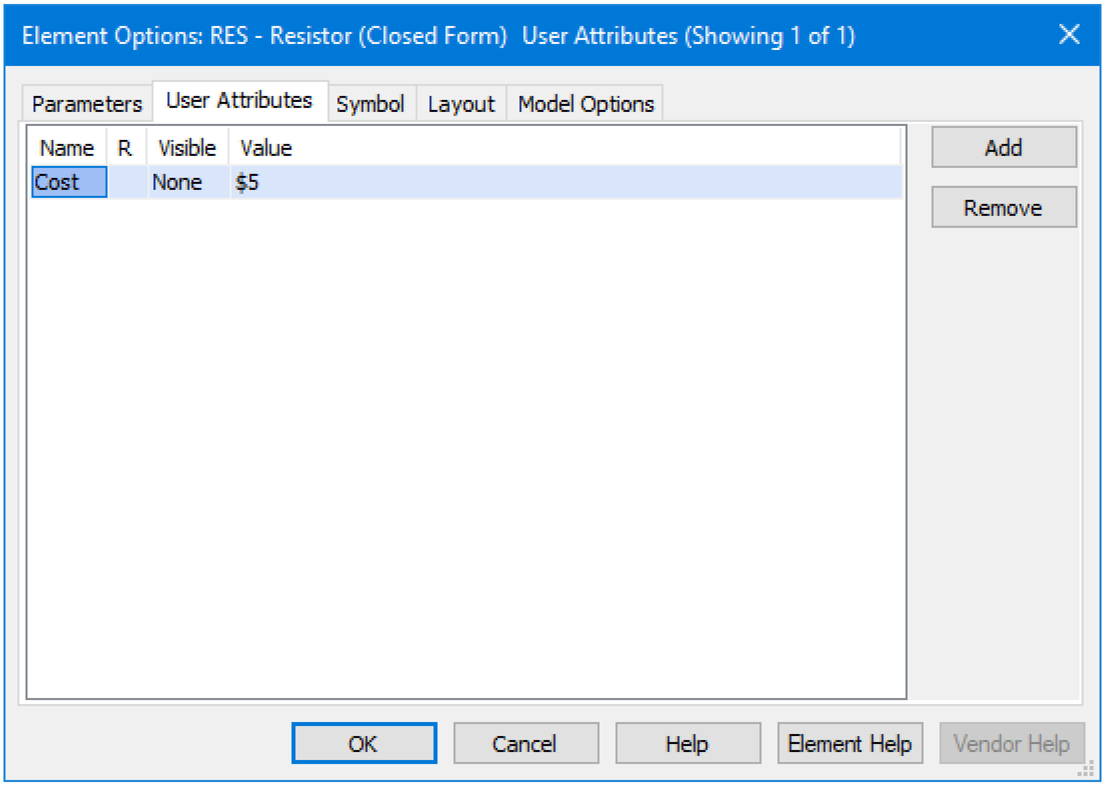
4.13. Exporting Schematics and System Diagrams

NOTE: Importing a schematic or system diagram into another project using the **Import Project** command is generally more efficient than using the following export process. See "Importing a Project " for more information.

To export a schematic or system diagram, right-click its node in the Project Browser and choose **Export Schematic** or **Export System Diagram**. In the dialog box that displays, specify the name and location for the exported file.

4.14. Adding User Attributes to Schematics and System Diagrams

The Element Options dialog boxes for Microwave Office elements and VSS models include a **User Attributes** tab that allows you to assign user-defined properties such as cost, weight, power, and current. The following figure shows the user attribute "Cost" with its value.



To add a new user attribute:

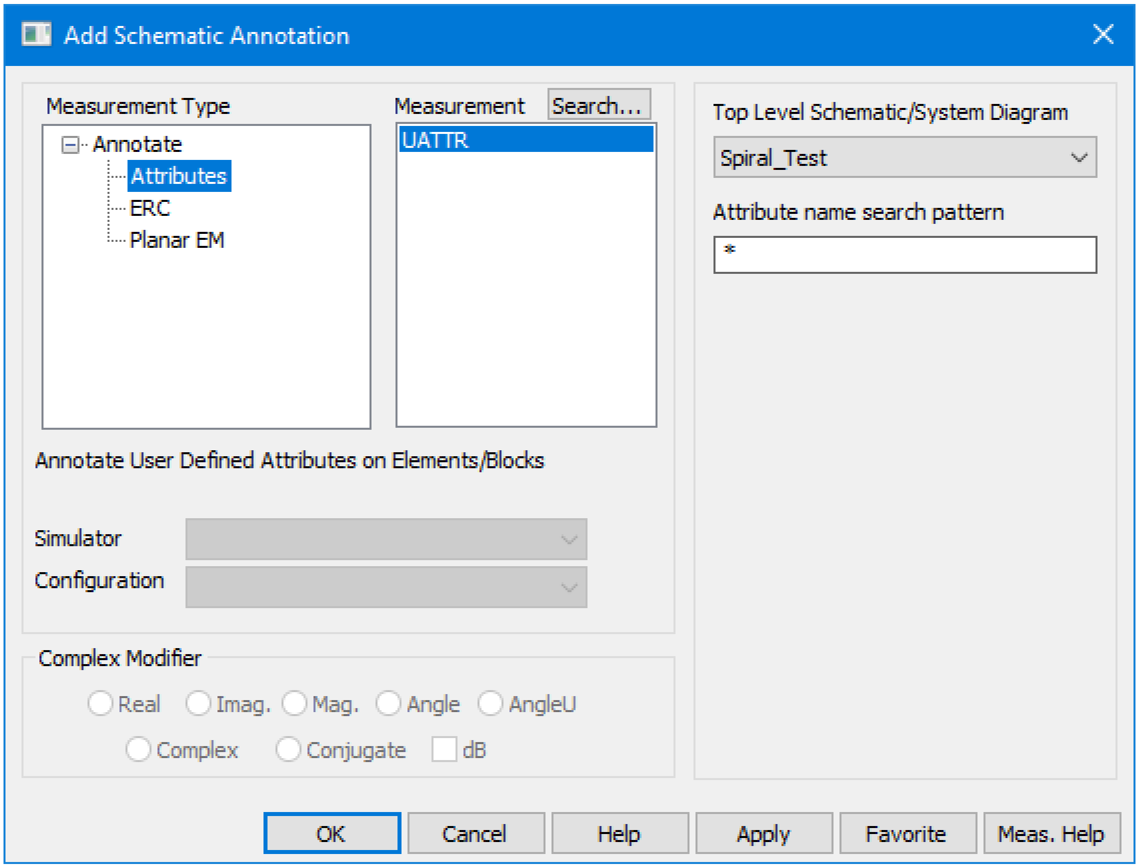
1. Double-click a model in a schematic or system diagram to open the Element Options dialog box, then click the **User Attributes** tab.
2. Click the **Add** button and type a **Name** and a **Value**. Both fields are string data type. The **Visible** drop-down list controls whether or not the attribute displays in the schematic or system diagram.

To delete a user attribute:

1. Double-click a model in a schematic or system diagram to open the Element Options dialog box, then click the **User Attributes** tab.
2. Select the user attribute you want to delete, then click the **Remove** button.

User attributes display alphabetically in a schematic or system diagram. To display the user attributes with the **Visible** option set to "None", you need to use the "Annotate User Defined Attributes on Elements/Blocks: UATTR" UATTR annotation:

1. Right-click the schematic or system diagram node in the Project Browser and choose **Add Annotation** to display the Add Schematic/System Diagram Annotation dialog box.
2. Select **Annotate > Attributes** as the **Measurement Type**, then select UATTR as the **Measurement**.
3. In **Attribute name search pattern**, specify which user attributes to display by entering a search pattern that filters attributes by name. For example, type "*" to display all user attributes, or type "C*" to display only the attributes whose names start with the letter "C".



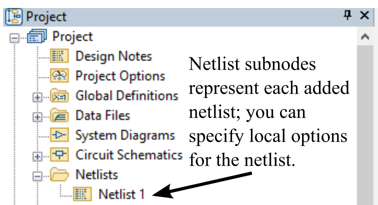
NOTE: You can add or modify user attributes through XML and the API. For XML access, see "Adding User Attributes in XML Files", and for API access see "Accessing the Element Properties Collection".

Chapter 5. Netlists

While a schematic is a graphical representation of an electrical circuit, a netlist is a text-based description of a circuit expressed in a standard format. In Cadence® Microwave Office® software, netlists are used to define subcircuits, which you can place in higher level schematics as SUBCKT blocks. These netlists can be expressed in the Cadence APLAC® HB , HSPICE®, or Spectre® simulator format. Microwave Office software can also translate Touchstone® and PSpice® netlists to Cadence format as they are imported.

5.1. Netlists in the Project Browser

The **Netlists** node in the Project Browser contains a subnode for each netlist that you create or import into the Microwave Office program for that project. The following figure shows a Netlist subnode.



5.2. Creating a Netlist

To create a new netlist:

1. Right-click **Netlists** in the Project Browser and choose **New Netlist**, or choose **Project > Add Netlist > New Netlist**.

   The New Netlist dialog box displays.

2. Enter a name for the netlist and select a netlist type, then click **Create**. (See the following sections for information about netlist types.)

   An empty netlist window opens in the workspace, and the Project Browser displays the new netlist as a node under **Netlists**. For information on how to add data to a new netlist, see “Adding Data To and Editing a Netlist”. Note that any netlist created within the Microwave Office program adheres to the AWR netlist format.

5.3. Importing or Linking to a Netlist

You can import electrical models into the Cadence AWR Design Environment® platform as third-party (HSPICE, Spectre, PSpice, and APLAC) simulator netlists, edit them as needed, and use them as SUBCKT blocks in schematics. The desired model must be a subcircuit definition in the netlist.

To import an existing netlist:

1. Right-click **Circuit Schematics** in the Project Browser and choose **Import Netlist**, or choose **Project > Add Netlist > Import Netlist**.

   The Browse For File dialog box displays.

2. Specify the type of file you want to import and locate the desired netlist. You can choose files of the following type:

File Type	Description
APLAC Files (native)	Netlist in APLAC format. Completely supports APLAC simulator syntax. Measurements must use an APLAC simulator in the AWR Design Environment platform. Files of this type have a *.lib* extension.
AWR Netlist Files	Netlist in AWR format. Files of this type have a *.net* extension.
HSPICE Files	Netlist in HSPICE® format is parsed so measurements can use any simulator. Supports commonly used HSPICE syntax. Files of this type have a *.sp* extension.
PSpice Files	PSpice file is translated into AWR format. Files of this type have a *.cir* extension.
Touchstone Files	Touchstone file is translated into AWR format. Files of this type have a *.ckt* extension.
Spectre Netlist Files	Netlist in Spectre™ format is parsed so measurements can use any simulator. Supports commonly used Spectre syntax. Files of this type have a *.scs* extension.

3. Click **Open** to copy the file and include it in the project. A netlist window opens in the workspace, and the Project Browser displays the imported netlist as a node under **Netlists**.

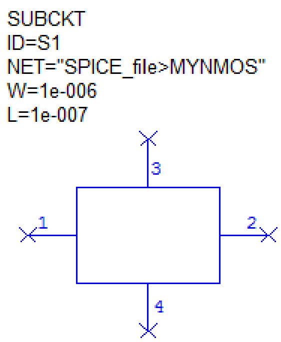
Alternatively, you may want to access a netlist without copying it into the project. To link to a netlist, right-click **Netlists** in the Project Browser and choose **Link To Netlist**. The Browse For File dialog box displays. Locate the desired netlist, and click **Open** to make the file part of the project. A netlist window opens in the workspace, and the Project Browser displays the linked netlist as a node under **Netlists**.

NOTE: When you link to a netlist, that file must always be available for the project to read.

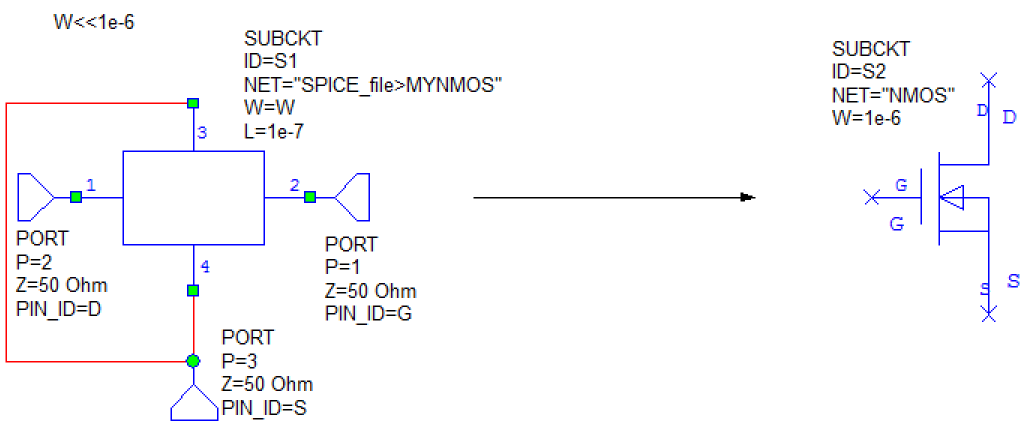
When you add an imported netlist model to a schematic as a subcircuit, the node names on the SUBCKT block correspond to the nodes in the netlist line(s) that define the subcircuit. For example, if an imported HSPICE subcircuit netlist starts with ".subckt MyModel 4 5 6", the SUBCKT block for MyModel has node numbers 4, 5, and 6.

A subcircuit definition in a netlist can include a list of parameters (usually after the terminal node list). When this subcircuit is placed in a schematic, the parameters display with the SUBCKT symbol. The following is an HSPICE netlist example. The symbol that displays when you place it in a schematic is shown in the following figure.

```
.SUBCKT MyNMOS 1 2 3 4 W=1u L=0.1u

M1 1 2 3 4 mname W='W' L='L'

.MODEL mname NMOS (LEVEL=3 …

…

.ENDS
```



Placing the imported netlist model as a single subcircuit in an otherwise empty, and similarly named schematic (as follows, on the left) provides the flexibility illustrated by the following figure and description:



- Default symbol assignment: You can assign a specific symbol to this schematic so it displays with that symbol instead of the generic default "box" symbol when you place it in another schematic as a subcircuit (on the right, above).
- Node count reduction and re-ordering: If the model has four nodes, but you only need to use it as a 3-port (for example, with two of the four nodes connected together), you can edit this schematic as necessary, and add ports to the desired three nodes. You can also use the ports in this schematic to re-order the nodes as necessary, to match the order of pins on the symbol you want to use.
- Parameter count reduction, and alternate default values: If the model has many parameters, but you only want to access a few of them and leave the rest at default values, you can choose the parameters to expose and set different default values for them.
- Pin name assignment: Use the port PIN_ID parameter to assign names to the nodes, so that the SUBCKT blocks that refer to it are correctly annotated.
- Indirect reference: If this model is used in other schematics and you find a better model (schematic or netlist), you only need to edit this schematic to update all instances of it, rather than editing each SUBCKT block.

5.3.1. Imported Netlist Types

The following sections provide import information for individual netlist types.

5.3.1.1. HSPICE Netlist Files (*.sp) and Spectre Netlist Files (*.scs)

- These files can contain multiple subcircuit definitions and references to external files using the corresponding simulator's library reference syntax.
- Text in the file is imported without any modifications, but is parsed during import, and before simulations.
- Any messages from the parser display in the Status Window.
- If the file is successfully parsed, schematics containing subcircuit(s) from it can be analyzed by any simulator (measurements on those schematics can specify any simulator).
- When placing a SUBCKT block or editing its NET parameter, specific subcircuit definitions in the netlist are identified as *filename>subname*; where *filename* is the name of the netlist (without extension), and *subname* is the name of the desired subcircuit, as defined in the file.
- Most SPICE netlist files are written using standard SPICE 2G6 syntax, so you can import them as HSPICE netlist files. To do so, change the file extension to *.sp*, import the netlist, and check for any parsing messages in the Status Window.

5.3.1.2. APLAC Netlist Files (native) (*.lib)

- When a netlist is imported as "native", it can only be analyzed by the simulator for which it was written. For example, if a subcircuit model from a native APLAC netlist is included in schematic "A" or the hierarchy below "A", then measurements with "A" as their Data Source must specify an APLAC simulator.
- Native netlists are not restricted by parser limitations. They can include any syntax the simulator supports.
- A native netlist represents only one subcircuit-- the top level subcircuit if there is a hierarchy with multiple subcircuits. Ideally, the top level subcircuit is the first subcircuit defined in the file.
- The Microwave Office program lightly parses the netlist to determine the name of the subcircuit, the number of nodes, and any parameters and their default values-- just enough information to place it as a SUBCKT block in a schematic.
- For simulation, the main circuit netlist written by the Microwave Office program includes any native netlist's subcircuit definitions without modification (native netlists are fed directly to the circuit simulator). Microwave Office software only ensures that instances referring to it have the proper connections (node numbers) and parameters.

5.3.1.3. PSpice Files (*.cir) and Touchstone Files (*.ckt)

- These netlists are translated to AWR format during import.
- After import, any simulator can analyze the translated netlist.
- To edit the netlist in the original syntax, do so before importing it. Remove any previously imported version from the project and import the newly edited version.
- Translator messages display in the Status window after import. Some information in the form of comments may also be included in the translated/imported netlist.

5.3.1.4. AWR Netlist Files (*.net)

- If you must save your circuit descriptions in text format, you can export and import netlists in AWR format. Otherwise this is counterproductive.
- Instead of AWR netlists, you can easily export and import schematics, which are superior descriptions, can include much more information (for example, a default symbol and a layout), and are much easier to edit properly.

5.3.2. Importing Transistor Model Netlists and Swapping Nodes

SPICE and Spectre netlist subcircuits for transistor models usually order the nodes with the drain or collector first, and the gate or base second. Symbols in the Microwave Office program order the nodes as they are in 2-port S-parameters: gate or base first, drain or collector second. (The remaining nodes, source/emitter and the optional bulk node, are in the same order.) When you import a netlist that contains a subcircuit for a transistor model, you need to reconcile this potential difference before using one of the provided symbols for that subcircuit. There are a few ways to do so:

- As described previously, place the netlist as a subcircuit in an intermediate schematic. Attach a PORT element with parameter P=1 to the node corresponding to the gate or base, and one with P=2 to the drain or collector node (and ports 3 and 4 as appropriate). Edit the schematic options to select the desired symbol and then place the schematic as a subcircuit in schematics in which you want to use the transistor.
- Some netlist types allow you to automatically swap the first two nodes when you import a 3- or 4-pin subcircuit netlist. If not, you can manually edit the netlist before importing to swap the first two nodes in the ".SUBCKT..." line of the SPICE netlist.
- You can create a custom symbol for the model you are importing, with pins ordered so they match the model.

5.3.3. Importing a SPICE Netlist

Many commercial versions of SPICE exist, and some of them have custom syntax added to the standard from which they were derived (usually SPICE2G6 or SPICE3). The file extensions for commercial SPICE simulator netlists are not unique. If a netlist does not include a comment that identifies the simulator for which it is intended, you need to check the syntax for characteristics unique to each simulator to determine it.

To import a SPICE model:

1. If the netlist does not include a subcircuit definition (.SUBCKT…), but only includes a model definition (.MODEL…), then you must either edit it to “wrap” it in a .SUBCKT, or import it using the "Circuit_Model_Parameter_Read" script.
   - To import the .MODEL using the script, ensure that there is only one .MODEL in the file. Edit the file and replace the second word after “.MODEL” with the corresponding element name in the Microwave Office program. For example, if the model is a level 3 N-channel MOSFET model, and you want it to appear as a 4-pin FET in the schematic (with a visible bulk pin), then change:

     ```
     .MODEL mname NMOS (LEVEL=3 …
     ```

     to:

     ```
     .MODEL mname MOSN3_4A (LEVEL=3 …
     ```

     Then, choose **Scripts > Models > Circuit_Model_Parameter_Read** to run the script that reads the model.
   - To wrap the same example model in a .SUBCKT so you can import it as a netlist, edit it as follows:

     ```
     .SUBCKT MyNMOS G D S B

     M1 D G S B mname

     .MODEL mname NMOS (LEVEL=3 …

     …

     .ENDS
     ```

     To access the FET width and length parameters when you place it on the schematic, edit it as follows:

```
.SUBCKT MyNMOS G D S B W=1u L=0.1u
M1 D G S B mname W='W' L='L'
.MODEL mname NMOS (LEVEL=3 …
…
.ENDS
```

Ensure that the file name extension is *.sp*, and import it as an HSPICE file.

2. If the netlist is a subcircuit definition intended for use with HSPICE or PSpice, ensure that the netlist file name extension is *.sp* or *.cir*, respectively, and import it. If you do not know the target simulator for the netlist, view the netlist file using a text editor and determine if the comments identify the target as HSPICE or PSpice.
3. If the target is not HSPICE or PSpice, or if it is unknown, change the netlist file extension to *.sp* and import it as an HSPICE netlist file. After import, check the Status Window for messages. If the netlist format is SPICE2G6 compatible, it should import without errors.
4. Change the file extension to *.cir* and import it as a PSpice netlist.
5. Check for errors or warnings in the Status Window. If there are none, place the imported netlist in a schematic as a subcircuit, and verify the model's behavior using a simple test circuit (for example, a transistor's IV curves or an operational amplifier's bandwidth in a simple feedback setup).

5.3.3.1. PSpice Netlist Import Details

PSpice syntax includes many extensions to SPICE2G6 that cannot be translated during import. Fortunately, most models do not include the extended PSpice syntax. The following list includes some of the most common problematic syntax along with information on how to modify the netlist so that you can import it, if possible. These are errors observed in some vendor-distributed PSpice models, and proper PSpice syntax that cannot be imported.

- Remove extra periods (".") from command lines. (This is not standard PSpice syntax, but is used on some PSpice models distributed by vendors.)

  Original netlist:

  ```
  ..SUBCKT xx1001 10 20 30 40 50
  ```

  Fixed netlist:

  ```
  .SUBCKT xx1001 10 20 30 40 50
  ```

- Look for comments that do not have an asterisk ("*"), the standard SPICE comment symbol, as the first character in the line. PSpice also uses the semicolon (";") to mark the rest of the line as a comment.

  Original netlist:

  ```
  * xx1001 SPICE Macro-model rev A; 01/01/01
  Copyright 2002 by Company XYZ, Inc.
  .SUBCKT ABC 1 2 3 This is a comment
  ```

  Fixed netlist:

  ```
  * xx1001 SPICE Macro-model rev A; 01/01/01
  ```

```
* Copyright 2002 by Company XYZ, Inc.
.SUBCKT ABC 1 2 3 ; This is a comment
```

- Resistors, capacitors, and inductors with .MODEL statements may not translate properly. Remove the model name from the element instance, multiply the value in the instance by the R, C, or L parameter value specified in the .MODEL statement (default is 1), and comment out the .MODEL statement.

  Original netlist:

```
Rxx 1 2 rmodel 10
Cxx 2 3 cmodel 2p
Lxx 3 4 lmodel 30
.MODEL rmodel RES (T_ABS=-273)
.MODEL cmodel CAP (C=10 …)
.MODEL lmodel IND (L=1n …)
```

  Fixed netlist:

```
Rxx 1 2 10 ; R was not in .MODEL, and defaults to 1
Cxx 2 3 20p
Lxx 3 4 30n
*.MODEL rmodel RES (T_ABS=-273)
*.MODEL cmodel CAP (C=10 …)
*.MODEL lmodel IND (L=1n …)
```

- PSpice has a voltage-controlled switch (S element, VSWITCH model) that you cannot import into the AWR Design Environment platform. For alternatives, contact Technical Support for AWR Products.

  Original netlist:

```
SS1 1 2 3 4 smodel
…
.MODEL smodel VSWITCH(Voff=0 Von=1 Roff=1e7 Ron=1e-3)
```

- Controlled sources (E and G elements) with VALUE, TABLE, LAPLACE, FREQ, or CHEBYSHEV keywords. These elements implement the extensive PSpice analog behavioral modeling capabilities, and you cannot import them directly.

5.3.3.2. PSpice and Berkeley SPICE MOSFET Model Level 3

This MOSFET model dates back to the old SPICE2G6 from U.C. Berkeley. The default values for some of its parameters are not the same in the Microwave Office program (or HSPICE) as they are in Berkeley SPICE or PSpice. This is not a problem when the model parameter values are specified; however, SPICE "macro-models" (subcircuit netlists) often include near ideal models with very few of their parameters specified, and the rest left as default. Unfortunately, the intended SPICE simulator cannot be automatically identified. Before you import a SPICE netlist that is intended for use in PSpice or other Berkeley SPICE-compatible simulator, check the netlist for NMOS or PMOS models with the parameter setting LEVEL=3.

```
.MODEL mname NMOS(... LEVEL=3...)
```

or

```
.MODEL mname PMOS(... LEVEL=3...)
```

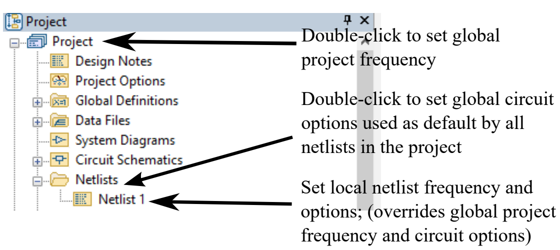
Check each you find for the following parameters, and follow the instructions:

- U0: If not specified, set U0=600.
- GAMMA: If not specified, set GAMMA=0.002 (small, non-zero value).
- NSUB: If not specified, set KAPPA=0 (this is not an error, do not set NSUB).
- PHI: If not specified, set PHI=0.6V.
- CJ: If not specified, set CJ=0.
- LD: If not specified, set LD=0.
- ETA: If specified and non-zero, multiply the value of ETA by 815/814.
- XJ: If specified and non-zero, XJ should never be smaller than 0.05um.

5.4. Specifying Netlist Options

Netlist options include settings that control how the harmonic balance simulator performs its calculations, what type of solver is applied to linear simulations, and what format the netlist is expressed in, as well as simulation frequency.

You can configure netlist options for a particular netlist via the netlist subnode, or you can use the default options set for all netlists contained within the project. These choices are shown in the following figure.



5.4.1. Configuring Global Circuit Options

To configure global circuit options:

1. Follow the steps in "Configuring Global Circuit Options " to configure global circuit options. The options that apply to netlists are located under the Harmonic Balance Options and Circuit Solver Options sections of the appropriate simulator tab on the Circuit Options dialog box. If you don't configure global circuit options, the default global circuit options are used.
2. Follow the steps in "Configuring Global Project Frequency" to configure global frequency. If you don't configure global frequency, the default global frequency is used.

5.4.2. Configuring Local Netlist Options and Frequency

To configure local netlist options:

1. Right-click the desired netlist and choose **Options**. The Options dialog box displays.
2. Click the **Frequencies** tab and specify the desired local simulation frequency. See "Options Dialog Box: Frequencies Tab" for more information about this dialog box.
3. The options that apply to harmonic balance are located under the Harmonic Balance Options section of the appropriate simulator tab on the Circuit Options dialog box.
4. The options that apply to circuit solvers are located under the Circuit Solver Options section of the **AWR Sim** tab on the Circuit Options dialog box.
5. Click **OK**.

5.5. Adding Data To and Editing a Netlist

When you create a new netlist, an empty netlist window opens to allow you to type a text-based description of a schematic that adheres to the specified netlist format. Netlist data is arranged in blocks in a particular order, where each block defines a different attribute of an element, such as units, equations, or element connections. The AWR netlist format (the blocks, their attributes and allowed values, etc.) is described in "AWR Netlist Format".

To add data to or edit a netlist:

1. Create a new netlist as described in "Creating a Netlist", or open an existing netlist by double-clicking the netlist in the Project Browser. The netlist window opens in the workspace.
2. Create the blocks required to define the circuit in the order described in "AWR Netlist Format". Note that an AWR netlist must at minimum have a `CKT` block; other blocks are optional.
3. Fill in the attributes and attribute values for each block. See "AWR Netlist Format".
4. To save the netlist and close the netlist window, click the **X** in the upper right corner of the netlist window.

5.6. Copying a Netlist

To copy a netlist, drag the individual netlist node to the **Netlists** node in the Project Browser. A subnode named "netlistname_1" is created for the first copy and incremented by one (_2, _3 and so on) for each additional copy.

5.7. Renaming a Netlist

To rename a netlist, right-click the netlist node under **Netlists** in the Project Browser and choose **Rename Netlist**. The Rename Netlist dialog box displays. See "Rename Netlist Dialog Box" for more information about this dialog box.

5.8. Exporting a Netlist

NOTE: Importing a netlist into another project using the **Import Project** command is generally more efficient than using the following export process. See "Importing a Project " for more information.

To export a netlist, right-click the netlist in the Project Browser and choose **Export Netlist**. The Export Netlist dialog box displays. Select a **Save in** directory, specify a file name, and click **Save**. Note that netlists created within the Microwave Office program have a *.net* extension.

5.9. AWR Netlist Format

In the AWR netlist format, data is expressed using a series of "blocks" that must be arranged in a strict order. Each block provides particular information about the circuit being defined, such as the declaration of elements that comprise the circuit or the dimensions used by the circuit.

5.9.1. Netlist Blocks

The netlist blocks are described in the following table, listed in the order in which they must display in the netlist file:

Netlist Block	Description
DIM	Optional. Not case-sensitive. Defines the dimensions of the units used by the circuit. Default values are: resistance in ohms, length in meters.
VAR	Optional. Case-sensitive. Defines the variables used by the elements (and possibly by the equations) in the circuit.
EQN	Optional. Case-sensitive. Defines the equations used by the elements in the circuit.
CKT	Mandatory. Case-sensitive. Defines the elements that comprise the circuit and the connections between them.

5.9.1.1. DIM Block

The following table shows an example of a `DIM` block, which defines the dimensions of the units used by the circuit. The unit is not case-sensitive.

```
DIM
LNG mil
RES OH
ANG rad
FREQ GHz
```

The `DIM` block keywords and allowed values are described in the following table.

Keyword	Allowed Values	Description
FREQ	Hz, kHz, MHz, GHz	Frequency
RES	OH, kOH, MOH	Resistance
COND	/OH, /kOH, /MOH	Conductance
IND	pH, nH, uH, mH, H	Inductance
CAP	fF, pF, nF, uF, mF, F	Capacitance
LNG	um, mm, cm, m	Metric length
LNG	mil, in	English length
TIME	ps, ns, us, ms, sec	Time
ANG	rad, deg	Angle
VOL	uV, mV, V, kV	Voltage
CUR	pA, nA, uA, mA, A	Current

5.9.1.2. VAR Block

The following is an example of a VAR block, which defines the variables used by the elements and equations in this circuit.

```
VAR
a = 10
c \ -3
d # -1 0 1
```

This block is composed of a series of expressions consisting of a variable name, followed by an operator, followed by numeric data.

The variable name must follow these rules:

- The variable name may contain the characters A-Z, a-z, 0-9, and "_" (underscore), with no spaces.
- The first character of the variable name may not be a digit.
- The variable name is case-sensitive.

The VAR block operators and allowed data are shown in the following table.

Operator	Data	Description
=	single number	Specifies data that is not optimizable.
\	single number	Specifies data that is allowed to be optimized.
#	3 numbers	Specifies data that is allowed to be optimized within the specified range. The first number is the low value, the middle is the nominal value, and the third is the high value.

5.9.1.3. EQN Block

The following is an example of an EQN block, which defines the equations used by the elements and other equations in this circuit.

```
EQN
a = 5
e = 10
b = { 1 + a*e }
g = { sin( _PI ) }
```

This block is composed of a series of expressions consisting of an equation name, followed by an "=" operator, followed by a mathematical expression.

The equation name has the following restrictions:

- The first character of the variable name must be a letter.
- The equation name may contain the characters A-Z, a-z, 0-9, and "_" (underscore), with no spaces.
- The equation name is case-sensitive.

The mathematical expression has the following restrictions:

- If the expression has more than one component, then the entire expression must be enclosed in braces.
- A set of built-in functions such as `sin(x)` can be used. The built-in functions are: `sin(x)`, `cos(x)`, `tan(x)`, `sinh(x)`, `cosh(x)`, `tanh(x)`, `arcsin(x)`, `arccos(x)`, `arctan(x)`, `exp(x)`, `log(x)`, `log10(x)`, `sqrt(x)`.

5.9.1.4. CKT Block

The following shows the format of the `CKT` block, which defines the elements that comprise the circuit and the connections between them.

```
CKT
element_name node_list parameter_list
element_name node_list parameter_list
element_name node_list parameter_list
DEFnP port_list circuit_name circuit_parameter_list
```

An example of the `CKT` block is shown here

```
CKT
RES 1 2 R \ 10
TLIN 2 3 Z0 = 50 \
RES 2 0 R = R
EL # 100 200 500
SUBCKT 2 4 NET = filter
GND 3
DEF2P 1 4 circuit1 R=100
```

The `CKT` block keywords and allowed values are described in the following table.

Keyword	Allowed Values	Description
`element_name`	Recognizes all elements in the Elements Browser.	Name of the element.
`node_list`	0-9	A series of numbers, each of which identifies one of the element's nodes.
`parameter_list`	See `EQN` and `VAR` descriptions for valid expressions. Expressions must be separated by spaces, and may continue to a new line by using a backslash.	A series of expressions that assign values to the element's parameters. A parameter is assigned a value in the same manner as the equations in the `EQN` block, or they can be set up for optimization using the same format as the variables in the `VAR` block. Each parameter listed here must match a parameter name in the element definition specified in the Elements Browser, and they must be in the same order; any missing parameters are assigned default values.
`DEFnP`		Defines the number of external ports for the element, where "`n`" is the number of external ports.

Keyword	Allowed Values	Description
`port_list`	1-9	A series of numbers, each of which identifies a node that is an external port. The number of items in the list must match the `n` in `DEFnP`.
`circuit_name`	String of alphanumeric characters.	The name of this `CKT` block. Must begin with a letter.
`circuit_parameter_list`	Variable name, followed by equal sign and default value.	Optional list of parameters to be passed into the circuit.

5.9.2. Netlist Example

This example describes a 2-port circuit that uses a subcircuit element to connect a data source defined elsewhere in the project.

NOTE: Comments begin with an exclamation point and end with the end of the current line.

```
! unit dimensions
DIM
  RES kOH   ! in kOhm
  IND nH   ! in nH
! global variables
VAR
  a=10
  d # -1 0 1   ! range
! global equations
EQN
  b={1+a}
! circuit block
CKT
  RES 1 2 R \ 10  ! 10 kOhm, optimizable
  IND 2 3 L={b*2}  ! evaluates to 22 nH
  SUBCKT 2 4 NET=filter
  GND 3
```

5.10. Touchstone File Import Utility

This section provides a brief description of the Touchstone File Import utility used with Microwave Office software. The import utility currently handles most of the elements in Touchstone. Library functionality is ignored in this version of the utility. Some of the elements, such as FREQ and OUT are not translated into the Microwave Office program. It is necessary to manually set up the new project to properly duplicate the functionality of the Touchstone file. A table at the end of this section describes which Touchstone elements and parameters the translator supports.

For information about importing Touchstone data for system designs, see “Linear Behavioral Model, Variable Port Count (Simulation-Based): LIN_S” and “LIN_F/LIN_F2 Versus LIN_S”.

The following example file demonstrates how the Microwave Office import utility formats the new netlist.

5.10.1. Example Touchstone File

The following is an example Touchstone file:

5.10.1.1. File format: Touchstone Circuit file

```
DIM

   FREQ  MHZ
   RES  OH
   COND  /OH
   IND  NH
   CAP  PF
   LNG  MIL
   TIME  NS
   ANG  DEG
   VOL  V
   CUR  MA
   PWR  DBM

VAR

   s1#1 1 3
   L1#20 23 26
   L2#15 17.5 21
   deltL#-2 -1 2
   deltaS#0 0.5 1
   w_sec1#2 3 4
   w_sec2#4 5.5 7
   vres_L#60 89 90
   deltw#-2 -0.5 2
   deltaS#0 0.5 1
   s2#1 1.5 3
   s3#1.5 2 4
   w3#7 7.5 9
   w_sec2=5.5
   hres_L3#15 17.5 21

EQN

   hres_L1=L1+deltL
   hres_L2=L2+deltL
   space_1=s1+deltaS
   w_sec3=w3+deltw
   hres_L3X2=2*hres_L3
   S2_delta=s2+deltaS
   S3_delta=s3+deltaS

CKT

MSUB Er=9.8 H=10 T=0.01
Rho=1 Rgh=.01
MLIN 2 7  W=w_sec1 L=hres_L1
MLIN 8 1  W=w_sec1 L=hres_L1
MLOC 9 W=w_sec1 L=0
MLOC 10 W=w_sec1 L=0
MSTEP 1 5  W1=w_sec1  W2=w_sec2
MLIN 5 6  W=w_sec2  L=hres_L2
MLOC 11 W=w_sec2  L=0
MLOC 12 W=w_sec2  L=0
MLIN 13 4 W=w_sec2  L=hres_L2
MBEND2 7 14  W=w_sec1
```

```
MBEND2 8 15  W=w_sec1
MBEND2 16 6  W=w_sec2
MBEND2 17 13  W=w_sec2
MSTEP 3 2  W1=10 W2=w_sec1
MCLIN 9 15 10 14  W=w_sec1  S=space_1 &    L= vres_L
MCLIN 16 12 17 11  W= w_sec2  S=space_1 & L= vres_L
DEF2P 3 4 quarter_1
MSTEP 3 4  W1= w_sec2  W2= w_sec3
MLIN 4 5  W= w_sec3  L= hres_L3
MLOC 6 W= w_sec3  L=0
MLOC 7 W= w_sec3  L=0
MLIN 1 2  W= w_sec3  L= hres_L3X2
MLOC 8 W= w_sec3  L=0
MLOC 9 W= w_sec3  L=0
MLIN 10 11  W= w_sec3  L= hres_L3
MBEND2 5 12  W= w_sec3
MBEND2 1 13  W= w_sec3
MBEND2 14 2  W= w_sec3
MBEND2 15 10  W= w_sec3
MCLIN 6 13 7 12  W= w_sec3  S= S2_delta & L= vres_L
MCLIN 14 8 15 9  W= w_sec3  S= S3_delta & L= vres_L
DEF2P 3 11 quarter_2
quarter_1 1 2
quarter_2 2 3
DEF2P 1 3 halfbpf
halfbpf 1 2
halfbpf 3 2
DEF2P 1 3 BPF2

FREQ

   SWEEP 6000 15000 100

OUTVAR

OUTEQN

 OUT

  BPF2 DB[S21] GR1
  BPF2 DB[S11] GR1
  BPF2 DB[S21] GR2
  BPF2 DB[S11] GR2A
GRID
  GR1 -50 0 5
  FREQ 6000 15000 100
  GR2 -5 0 0.5
  GR2A -25 0 2.5
  FREQ 7000 12000 100

OPT

  FREQ 8700 12700
  BPF2 DB[S11] <-20  10000
  FREQ 8700 12700
  BPF2 DB[S21] >-2  1000
```

The four imported netlists in the Microwave Office program are listed below.

5.10.1.2. Subcircuit: Quarter_1

```
DIM

   FREQ  MHZ
    RES  OH
   COND  /OH
    IND  NH
    CAP  PF
    LNG  MIL
   TIME  NS
    ANG  DEG
    VOL  V
    CUR  MA
    PWR  DBM

VAR

   S1#1 1 3
   L1#20 23 26
   L2#15 17.5 21
   DELTL#-2 -1 2
   DELTAS#0 0.5 1
   W_SEC1#2 3 4
   W_SEC2#4 5.5 7
   VRES_L#60 89 90
   DELTW#-2 -0.5 2
   DELTAS#0 0.5 1
   S2#1 1.5 3
   S3#1.5 2 4
   W3#7 7.5 9
   W_SEC2=5.5
   HRES_L3#15 17.5 21

EQN

   HRES_L1=L1 + DELTL
   HRES_L2=L2 + DELTL
   SPACE_1=S1 + DELTAS
   W_SEC3=W3 + DELTW
   HRES_L3X2=2 * HRES_L3
   S2_DELTA=S2 + DELTAS
   S3_DELTA=S3 + DELTAS

CKT

MSUB Er=9.8 H=10 T=0.01
Rho=1 Name="MSUB1"
MLIN 2 7 W=W_SEC1 L=HRES_L1
MSUB="MSUB1"
MLIN 8 1 W=W_SEC1 L=HRES_L1
MSUB="MSUB1"
MLOC 9 W=W_SEC1 L=0 MSUB="MSUB1"
MLOC 10 W=W_SEC1 L=0 MSUB="MSUB1"
MSTEP 1 5 W1=W_SEC1 W2=W_SEC2
MSUB="MSUB1"
MLIN 5 6 W=W_SEC2 L=HRES_L2
MSUB="MSUB1"
```

```
MLOC 11 W=W_SEC2 L=0 MSUB="MSUB1"
MLOC 12 W=W_SEC2 L=0 MSUB="MSUB1"
MLIN 13 4 W=W_SEC2 L=HRES_L2
MSUB="MSUB1"
MBEND2 7 14 W=W_SEC1 MSUB="MSUB1"
MBEND2 8 15 W=W_SEC1 MSUB="MSUB1"
MBEND2 16 6 W=W_SEC2 MSUB="MSUB1"
MBEND2 17 13 W=W_SEC2 MSUB="MSUB1"
MSTEP 3 2 W1=10 W2=W_SEC1 MSUB="MSUB1"
MCLIN 9 15 14 10 W=W_SEC1 S=SPACE_1
L=VRES_L MSUB="MSUB1"
MCLIN 16 12 11 17 W=W_SEC2 S=SPACE_1
L=VRES_L MSUB="MSUB1"
DEF2P 3 4 quarter_1
```

5.10.1.3. Subcircuit: Quarter_2

```
DIM

   FREQ  MHZ
    RES  OH
   COND  /OH
    IND  NH
    CAP  PF
    LNG  MIL
   TIME  NS
    ANG  DEG
    VOL  V
    CUR  MA
    PWR  DBM

VAR

   S1#1 1 3
   L1#20 23 26
   L2#15 17.5 21
   DELTL#-2 -1 2
   DELTAS#0 0.5 1
   W_SEC1#2 3 4
   W_SEC2#4 5.5 7
   VRES_L#60 89 90
   DELTW#-2 -0.5 2
   DELTAS#0 0.5 1
   S2#1 1.5 3
   S3#1.5 2 4
   W3#7 7.5 9
  W_SEC2=5.5
   HRES_L3#15 17.5 21

EQN

   HRES_L1=L1 + DELTL
   HRES_L2=L2 + DELTL
   SPACE_1=S1 + DELTAS
   W_SEC3=W3 + DELTW
   HRES_L3X2=2 * HRES_L3
   S2_DELTA=S2 + DELTAS
   S3_DELTA=S3 + DELTAS
```

```
CKT

MSUB Er=9.8 H=10 T=0.01
Rho=1 Name="MSUB1"
MSTEP 3 4 W1=W_SEC2 W2=W_SEC3
MSUB="MSUB1"
MLIN 4 5 W=W_SEC3 L=HRES_L3
MSUB="MSUB1"
MLOC 6 W=W_SEC3 L=0 MSUB="MSUB1"
MLOC 7 W=W_SEC3 L=0 MSUB="MSUB1"
MLIN 1 2 W=W_SEC3 L=HRES_L3X2
MSUB="MSUB1"
MLOC 8 W=W_SEC3 L=0 MSUB="MSUB1"
MLOC 9 W=W_SEC3 L=0 MSUB="MSUB1"
MLIN 10 11 W=W_SEC3 L=HRES_L3
MSUB="MSUB1"
MBEND2 5 12 W=W_SEC3 MSUB="MSUB1"
MBEND2 1 13 W=W_SEC3 MSUB="MSUB1"
MBEND2 14 2 W=W_SEC3 MSUB="MSUB1"
MBEND2 15 10 W=W_SEC3 MSUB="MSUB1"
MCLIN 6 13 12 7 W=W_SEC3 S=S2_DELTA
L=VRES_L MSUB="MSUB1"
MCLIN 14 8 9 15 W=W_SEC3 S=S3_DELTA
L=VRES_L MSUB="MSUB1"
DEF2P 3 11 quarter_2
```

5.10.1.4. Subcircuit: HALFBPF

```
DIM

   FREQ  MHZ
    RES  OH
   COND  /OH
    IND  NH
    CAP  PF
    LNG  MIL
   TIME  NS
    ANG  DEG
    VOL  V
    CUR  MA
    PWR  DBM

VAR

   S1#1 1 3
   L1#20 23 26
   L2#15 17.5 21
   DELTL#-2 -1 2
   DELTAS#0 0.5 1
   W_SEC1#2 3 4
   W_SEC2#4 5.5 7
   VRES_L#60 89 90
   DELTW#-2 -0.5 2
   DELTAS#0 0.5 1
   S2#1 1.5 3
   S3#1.5 2 4
   W3#7 7.5 9
```

```
   W_SEC2=5.5
   HRES_L3#15 17.5 21

EQN

   HRES_L1=L1 + DELTL
   HRES_L2=L2 + DELTL
   SPACE_1=S1 + DELTAS
   W_SEC3=W3 + DELTW
   HRES_L3X2=2 * HRES_L3
   S2_DELTA=S2 + DELTAS
   S3_DELTA=S3 + DELTAS

CKT

SUBCKT 1 2 NET=quarter_1
SUBCKT 2 3 NET=quarter_2
DEF2P 1 3 halfbpf
```

5.10.1.5. Subcircuit: BPF2

```
DIM

   FREQ  MHZ
    RES  OH
   COND  /OH
    IND  NH
    CAP  PF
    LNG  MIL
   TIME  NS
    ANG  DEG
    VOL  V
    CUR  MA
    PWR  DBM

VAR

   S1#1 1 3
   L1#20 23 26
   L2#15 17.5 21
   DELTL#-2 -1 2
   DELTAS#0 0.5 1
   W_SEC1#2 3 4
   W_SEC2#4 5.5 7
   VRES_L#60 89 90
   DELTW#-2 -0.5 2
   DELTAS#0 0.5 1
   S2#1 1.5 3
   S3#1.5 2 4
   W3#7 7.5 9
   W_SEC2=5.5
   HRES_L3#15 17.5 21

EQN

   HRES_L1=L1 + DELTL
   HRES_L2=L2 + DELTL
   SPACE_1=S1 + DELTAS
```

```
   W_SEC3=W3 + DELTW
   HRES_L3X2=2 * HRES_L3
   S2_DELTA=S2 + DELTAS
   S3_DELTA=S3 + DELTAS

CKT

SUBCKT 1 2 NET=halfbpf
SUBCKT 3 2 NET=halfbpf
DEF2P 1 3 BPF2
```

5.10.1.6. Microwave Office Project Setup after Touchstone Netlist Import

The following Touchstone blocks are excluded upon import into the Microwave Office program:

```
FREQ

OUTVAR

OUTEQN

OUT

GRID

OPT
```

The following table describes how each Touchstone block relates to an element in the Project Browser.

Touchstone block	Project Browser element
FREQ	Project Frequency
OUTEQN	Output Equations
GRID	Edit > Properties (with a graph active)
OUTVAR	Output Equations
OUT	Graph > Add Measurement
OPT	Optimizer Goals

After importing the Touchstone netlist you must go to the corresponding locations in the Project Browser to re-enter the data and parameters that are excluded from the import into the program.

```
Touchstone Block:FREQ

SWEEP 6000 15000 100
```

Project Options dialog box (double-click **Project Options** and then click the **Frequencies** tab on the Project Options dialog box).

```
Touchstone Block:OUT

  BPF2 DB[S21] GR1
  BPF2 DB[S11] GR1
  BPF2 DB[S21] GR2
  BPF2 DB[S11] GR2A
```

Add/Modify Measurement dialog box (right-click **Graphs**, choose **New Graph**, create and name a graph, right-click the new graph, then choose **Add New Measurement** to display the dialog box). Create a graph for each grid GR1 and GR2.

```
Touchstone Block:GRID

  GR1 -50 0 5
  FREQ 6000 15000 100
  GR2 -5 0 0.5
  GR2A -25 0 2.5
  FREQ 7000 12000 100
```

Graph Options dialog box. (Right-click in a graph window, then choose **Options**).

Repeat for grid 2.

```
Touchstone Block:OPT

  FREQ 8700 12700
  BPF2 DB[S11] <-20  10000
  FREQ 8700 12700
  BPF2 DB[S21] >-2  1000
```

New Optimization Goal dialog box. (Right-click **Optimizer Goals** in the Project Browser and choose **Add Optimizer Goal**).

5.10.1.7. Set Up Tunable and Optimizable Variables

Inspection of the Microwave Office program netlists shows that the equation and variable block are copied into each subcircuit's netlist. The files are imported in this manner to ensure that each subcircuit is a separate measurable circuit. Because of this import method, some of the variables and equations do not apply to the elements in the Microwave Office subcircuit netlist. This can cause confusion regarding which variables are being tuned or optimized in the various subcircuit netlist. You must delete the variables and equations that do not apply to a particular subcircuit. For the BPF2 circuit, all the variables and equations apply to subcircuits "quarter_1" and "quarter_2", so you can delete all variables in netlists "halfbpf" and "BPF2". The new netlists for both subcircuits are listed as follows.

5.10.1.8. Subcircuit BPF2

```
DIM

   FREQ  MHZ
    RES  OH
   COND  /OH
    IND  NH
    CAP  PF
    LNG  MIL
   TIME  NS
    VOL  V
    CUR  MA
    PWR  DBM

CKT

SUBCKT 1 2 NET=halfbpf
SUBCKT 3 2 NET=halfbpf
DEF2P 1 3 BPF2
```

5.10.1.9. Subcircuit - HALFBPF

```
DIM

   FREQ  MHZ
    RES  OH
   COND  /OH
    IND  NH
    CAP  PF
    LNG  MIL
   TIME  NS
    ANG  DEG
    VOL  V
    CUR  MA
    PWR  DBM

CKT

SUBCKT 1 2 NET=quarter_1
SUBCKT 2 3 NET=quarter_2
DEF2P 1 3 halfbpf
```

For netlists quarter_1 and quarter_2, you must inspect the netlist and delete the variables and equations that do not apply to the elements in the netlist. The new netlists are listed as follows.

5.10.1.10. Subcircuit Quarter_1

```
DIM

   FREQ  MHZ
    RES  OH
   COND  /OH
    IND  NH
    CAP  PF
    LNG  MIL
   TIME  NS
    ANG  DEG
    VOL  V
    CUR  MA
    PWR  DBM

VAR

   S1#1 1 3
   L1#20 23 26
   L2#15 17.5 21
   DELTL#-2 -1 2
   DELTAS#0 0.5 1
   W_SEC1#2 3 4
   W_SEC2#4 5.5 7
   VRES_L#60 89 90
   W_SEC2=5.5

EQN

   HRES_L1=L1 + DELTL
   HRES_L2=L2 + DELTL
   SPACE_1=S1 + DELTAS
```

```
CKT

MSUB Er=9.8 H=10 T=0.01
Rho=1 Name="MSUB1"
MLIN 2 7 W=W_SEC1 L=HRES_L1
MSUB="MSUB1"
MLIN 8 1 W=W_SEC1 L=HRES_L1
MSUB="MSUB1"
MLOC 9 W=W_SEC1 L=0 MSUB="MSUB1"
MLOC 10 W=W_SEC1 L=0 MSUB="MSUB1"
MSTEP 1 5 W1=W_SEC1 W2=W_SEC2
MSUB="MSUB1"
MLIN 5 6 W=W_SEC2 L=HRES_L2
MSUB="MSUB1"
MLOC 11 W=W_SEC2 L=0 MSUB="MSUB1"
MLOC 12 W=W_SEC2 L=0 MSUB="MSUB1"
MLIN 13 4 W=W_SEC2 L=HRES_L2
MSUB="MSUB1"
MBEND2 7 14 W=W_SEC1 MSUB="MSUB1"
MBEND2 8 15 W=W_SEC1 MSUB="MSUB1"
MBEND2 16 6 W=W_SEC2 MSUB="MSUB1"
MBEND2 17 13 W=W_SEC2 MSUB="MSUB1"
MSTEP 3 2 W1=10 W2=W_SEC1 MSUB="MSUB1"
MCLIN 9 15 14 10 W=W_SEC1 S=SPACE_1
L=VRES_L MSUB="MSUB1"
MCLIN 16 12 11 17 W=W_SEC2 S=SPACE_1
L=VRES_L MSUB="MSUB1"
DEF2P 3 4 quarter_1
```

5.10.1.11. Subcircuit Quarter_2

```
DIM

   FREQ  MHZ
    RES  OH
   COND  /OH
    IND  NH
    CAP  PF
    LNG  MIL
   TIME  NS
    ANG  DEG
    VOL  V
    CUR  MA
    PWR  DBM

VAR

   VRES_L#60 89 90
   DELTW#-2 -0.5 2
   DELTAS#0 0.5 1
   S2#1 1.5 3
   S3#1.5 2 4
   W3#7 7.5 9
   W_SEC2=5.5
   HRES_L3#15 17.5 21

EQN
```

```
   W_SEC3=W3 + DELTW
 HRES_L3X2=2 * HRES_L3
 S2_DELTA=S2 + DELTAS
 S3_DELTA=S3 + DELTAS

CKT

MSUB Er=9.8 H=10 T=0.01
Rho=1 Name="MSUB1"
MSTEP 3 4 W1=W_SEC2 W2=W_SEC3
MSUB="MSUB1"
MLIN 4 5 W=W_SEC3 L=HRES_L3
MSUB="MSUB1"
MLOC 6 W=W_SEC3 L=0 MSUB="MSUB1"
MLOC 7 W=W_SEC3 L=0 MSUB="MSUB1"
MLIN 1 2 W=W_SEC3 L=HRES_L3X2
MSUB="MSUB1"
MLOC 8 W=W_SEC3 L=0 MSUB="MSUB1"
MLOC 9 W=W_SEC3 L=0 MSUB="MSUB1"
MLIN 10 11 W=W_SEC3 L=HRES_L3
MSUB="MSUB1"
MBEND2 5 12 W=W_SEC3 MSUB="MSUB1"
MBEND2 1 13 W=W_SEC3 MSUB="MSUB1"
MBEND2 14 2 W=W_SEC3 MSUB="MSUB1"
MBEND2 15 10 W=W_SEC3 MSUB="MSUB1"
MCLIN 6 13 12 7 W=W_SEC3 S=S2_DELTA
L=VRES_L MSUB="MSUB1"
MCLIN 14 8 9 15 W=W_SEC3 S=S3_DELTA
L=VRES_L MSUB="MSUB1"
DEF2P 3 11 quarter_2
```

The optimizable and tunable variables are displayed in the Variable Browser. The **Tune**, **Optimize**, and **Constrained** columns are selected for the variables that are tunable, optimizable, and constrained, respectively.

5.11. Touchstone File Translation Capabilities

The following tables provide a convenient comparison of Touchstone and AWR models, and show the status of AWR Design Environment platform support for common Touchstone models.

5.11.1. Touchstone/AWR Model Support

The following sections show tables that list supported models, models for future support, and unsupported models.

5.11.1.1. SUPPORTED MODELS

Touchstone	AWR	Notes
BIP	BIP	
BIPB	BIPB	
CAP	CAP	
CAPQ	CAPQ	AWR defines frequency response differently
CCCS	CCCS	

Touchstone	AWR	Notes
CCVS	CCVS	
CIND	CIND	
CIR3	CIRC	AWR has extra parameters
CLIN	CLIN	
CLINP	CLINP	
COAX	COAX	
COAXA	COAX	
DELAY	DELAY	AWR has extra parameters
FET	FET	AWR has an extra parameter
FET2	FET	
GAIN	GAIN	AWR has an extra parameter
GYR	GYR8R	
HYBPI	HYBPI	
IND	IND	
INDQ	INDQ	AWR defines frequency response differently
ISOLATOR	ISOL8R	AWR has extra parameters
MATCH	LOAD	AWR has an extra parameter
MBEND	MBEND	
MBEND2	MBEND2	
MBEND3	MBEND3	
MCLIN	MCLIN	AWR ignores layout parameters W1-W4
MCORN	MBENDR	
MCROS	MCROSS	
MCURVE	MCURVE	
MGAP	MGAP	
MLANG	MLANGE	AWR has variable # fingers and ignores layout parameter
MLANG6	MLANGE	AWR has variable # fingers and ignores layout parameter
MLANG8	MLANGE	AWR has variable # fingers and ignores layout parameter
MLEF	MLEF	
MLIN	MLIN	
MLOC	MLOC	
MLSC	MLSC	
MRSTUB	MRSTUB	AWR defines geometry differently
MSTEP	MSTEP	

Touchstone	AWR	Notes
MSUB	MSUB	Data structure. AWR has extra parameters and ignores RGH
MTEE	MTEE	
NEG1	NEG1	
NEG2	NEG2	
OPAMP	OPAMP	
PHASE	PHASE	
PIN	PIN	
PIN2	PIN2	
PLC	PLC	
PRC	PRC	
PRL	PRL	
PRLC	PRLC	
RES	RES	
S1Px	SUBCKT	
S2Px	SUBCKT	
S3Px	SUBCKT	
S4Px	SUBCKT	
SBEND	SBEND	
SCLIN	SCLIN	AWR ignores layout parameters W1-W4
SCROS	SCROSS	
SCURVE	SCURVE	
SHOR	GND	
SLC	SLC	
SLEF	SLEF	
SLIN	SLIN	
SLOC	SLOC	
SLSC	SLSC	
SMITER	SMITER	
SRC	SRC	
SRL	SRL	
SRLC	SRLC	
SSTEP	SSTEP	
SSUB	SSUB	Data structure. AWR has extra parameters.
STEE	STEE	
TLIN	TLIN	

Touchstone	AWR	Notes
TLIN4	TLIN4	
TLIN4A	TLIN4	
TLINP	TLINP	AWR uses different units for loss
TLINP4	TLINP4	AWR uses different units for loss
TLINP4A	TLINP4	AWR uses different units for loss
TLOC	TLOC2	
TLPOC	TLOCP2	AWR uses different units for loss
TLPSC	TLSCP2	AWR uses different units for loss
TLSC	TLSC2	
UNIT	SHORT	
VCCS	VCCS	
VCVS	VCVS	
XFER	XFMR	The AWR turns ratio is inverse
XFERA	XFMR	The AWR turns ratio is inverse

5.11.1.2. For FUTURE Support

Touchstone	AWR;	Notes
CAPP	CAPP	AWR has an extra parameter
CPW	CPW	
CPWG	CPWG	
DFET	DFET	
DIPOLE	DIPOLE	
INCOR2	INCOR2	
INOISE	INOISE	
INSQ	INOISE	
MACLIN	MACLIN	
MACLIN3	MACLIN3	
MCFIL	MCFIL	AWR ignores layout parameters W1-W2
MICAP1	MICAP1	
MICAP2	MICAP2	
MICAP3	MICAP3	
MICAP4	MICAP4	
MONOPOLE	MONOPOLE	
MRIND	MRIND	
MSLIT	MSLIT	
MTAPER	MTAPER	
MUC	MUC2_M	

Touchstone	AWR;	Notes
OPEN	OPEN	
PLCQ	PLCQ	AWR defines frequency response differently
RIBBON	RIBBON	
RIND	RIND	
RWG	RWG	
RWGINDF	RWGINDF	
RWGT	RWGT	
SBCLIN	SBCLIN	
SBEND2	SBEND2	
SLCQ	SLCQ	AWR defines frequency response differently
SLINO	SLINO	
SOCLIN	SOCLIN	
SPIND	SPIND	
SSCLIN	SSCLIN	
SSLIN	SSLIN	
SSSUB	SSSUB	Data structure
TFC	TFC	
TFR	TFR	
VIA	VIAT	Tapered-hole via
VNCOR2	VNCOR2	
WIRE	WIRE	
XFERP	XFERP	
XFERRUTH	XFERRUTH	

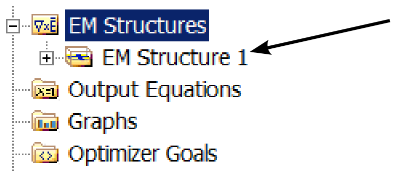
5.11.1.3. NOT SUPPORTED

Touchstone	AWR	Notes
FETN1		
FETN2		
FETN3		
FETN4		
FETN4A		
FETN5		
INCOR3		
MCOVER		Data structure
MWALL		Data structure

Touchstone	AWR	Notes
NPAR		Data structure
PERM	PERM	Data structure
RCLIN		
SIGMA	SIGMA	Data structure (defined in substrate definitions)
TAND	TAND	Data structure (defined in substrate definitions)
TEMP		Data structure (with no global temperature setting)
VINCOR		
VNCOR3		
VNOISE		
VNSQ		

Chapter 6. Electromagnetic Analysis

Electromagnetic (EM) structures are arbitrary multi-layered electrical structures. The **EM Structures** node in the Project Browser contains a subnode for each EM structure (also called EM document) in the project. The following figure shows the **EM Structures** node and its subnodes. Since the Cadence® AWR Design Environment® platform allows the integration of third-party electromagnetic solvers through EM Socket, each subnode representing an EM structure can have a separate EM solver associated with it.



EM structure subnodes represent each added EM structure. Right-click and choose Options to specify local EM options.

Parasitic extraction is a process where metal interconnects are simulated with a simulator to produce a model for the interconnect. Parasitic extraction and EM simulation are configured and used the same way; the only difference is the simulator used. EM simulation and parasitic extraction are performed in the same way.

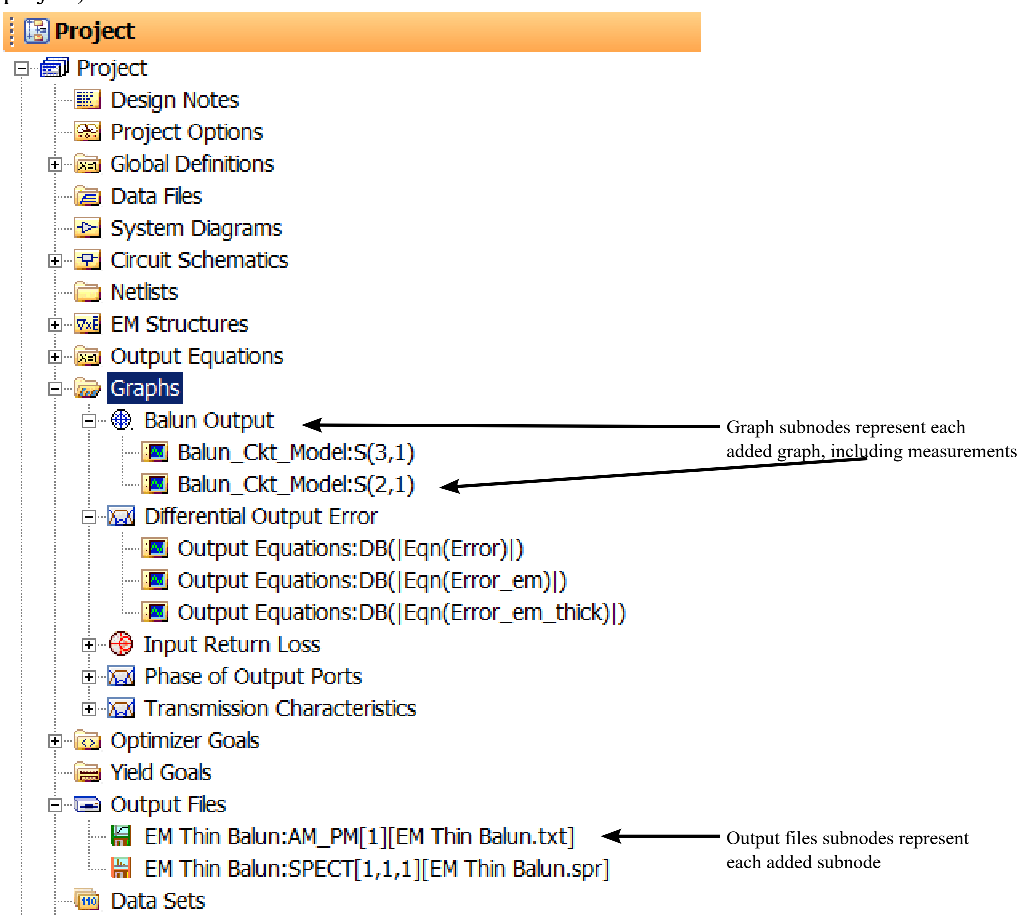
There are generally two methods you can use to create new EM structures. The first method is to manually create a new EM structure, add shapes (either by drawing or copying shapes from layouts), add ports, configure frequencies and options, and so on. This process involves performing the steps in the EM structure. The second method is to use Extraction flow, where EM structures are generated from schematic layouts.

See the AWR Design Environment Simulation and Analysis Guide for detailed information on EM simulators and methods.

Chapter 7. Graphs, Measurements, and Output Files

Before performing simulations, you need to specify the desired form of output for the results. The Cadence® AWR Design Environment® platform allows you to choose from a wide variety of results (measurements) to display in graphical form. As an alternative to displaying simulation results on graphs, you can also export Cadence Microwave Office® software results to output files in Touchstone®, SPICE, AM to AM, AM to PM, or spectrum data file format, and VSS™ (VSS) results to text file format.

The **Graphs** node in the Project Browser contains a subnode for each graph that you create for a project. The following figure shows the **Graphs** node and the **Output Files** node and its subnodes (for each output file that you create for that project).



7.1. Working with Graphs

The AWR Design Environment platform features extensive post-processing capabilities, allowing the display of computed data known as "measurements" on rectangular graphs, polar grids, Smith Charts, histograms, constellation graphs, tabular graphs, antenna plots, and 3D graphs. Highlights of the graphs features include:

- In the Microwave Office program, display of any port parameter (S, Y, Z, H, G or ABCD), VSWR, maximum gain, and stability.
- In the Microwave Office program, display of port impedance and propagation constant.

- In the Microwave Office program, display of box mode resonances for TE and TM modes.
- Display of the magnitude, angle, real, or imaginary component of any measurement using a dB or linear scale.
- Display of a live graph, schematic, system diagram, or layout (Microwave Office software only) within a graph.
- Reading of trace values from graphs using the data cursor.
- Changing the position and size of graphs and legends using click-and-drag operations.
- Zooming and panning to see small details.
- Changing a graph type and name using simple menu commands.
- Copying a graph (including all measurements and options) using simple menu commands.
- Copying measurements using simple menu commands.
- Copying a graph to the Design Notes window.
- Setting default graph options by graph type.
- Listing and modifying all measurements directly from graphs.
- Adding a drawn shape to a graph.

7.1.1. Creating a New Graph

You can create a new graph using any of the following methods:

- Right-click **Graphs** in the Project Browser and choose **New Graph**, or choose **Project > Add Graph**.

  The New Graph dialog box displays. See “New Graph Dialog Box” for more information about this dialog box.

  Enter a name for the graph, select the type of graph, and click **Create**. An empty graph window opens in the workspace, and the Project Browser displays the new graph as a subnode under **Graphs** in the Project Browser.

- Select an existing graph in the Project Browser, right-click and choose **Duplicate as**, and then select a graph type.

  A graph window opens in the workspace, and a graph of the selected type (including measurements and options) named "graphname 1" displays as a subnode under **Graphs** in the Project Browser. The graph name is incremented by one for each additional copy. When measurements differ between graph types the AWR Design Environment platform automatically applies the appropriate conversion.

- Select an existing graph and drag and drop it on the **Graphs** node in the Project Browser.

  A graph window opens in the workspace, and a duplicate graph (including measurements and options) named "graphname 1" displays as a subnode under **Graphs** in the Project Browser. The graph name is incremented by one for each additional copy.

  For more information about graph types see “Graph Types”. For information on how to specify which computed data (i.e., measurements) a graph displays, see “Adding a New Measurement”.

7.1.1.1. Using Default Graph Options

You can set and apply default graph options for individual graph types. To set default graph options, choose **Options > Default Graph Options**, and then choose a graph type. A Default Options dialog box displays with tabs for specifying all of the options associated with the particular graph type. These settings are used when you create a new graph.

To use the graph options from an existing graph as the default options for all graphs of that type, click **Save as Defaults** in the graph Options dialog box. The option settings on all of the dialog box tabs become the default settings for that graph type. Similarly, if you want to change an existing graph's options to the default options for that graph type, click **Reset to Defaults** in the graph Options dialog box.

7.1.1.2. Renaming a Graph

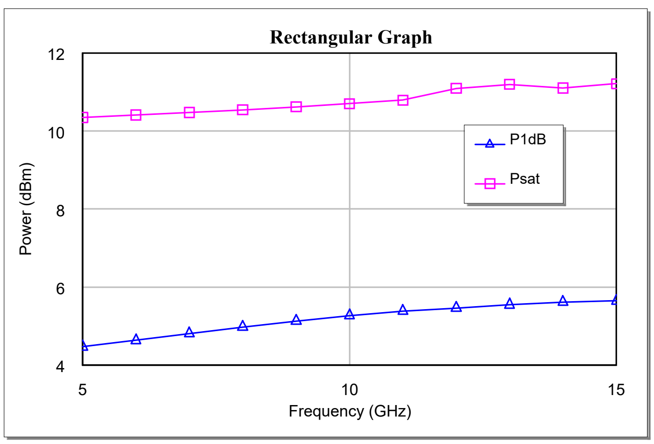
To rename a graph, right-click the graph in the Project Browser and choose **Rename Graph**. The Rename Output Document dialog box displays. Enter a new name for the graph, and then click **OK**.

7.1.2. Graph Types

The AWR Design Environment platform uses the following graph types for the display of measurements:

7.1.2.1. Rectangular Graphs

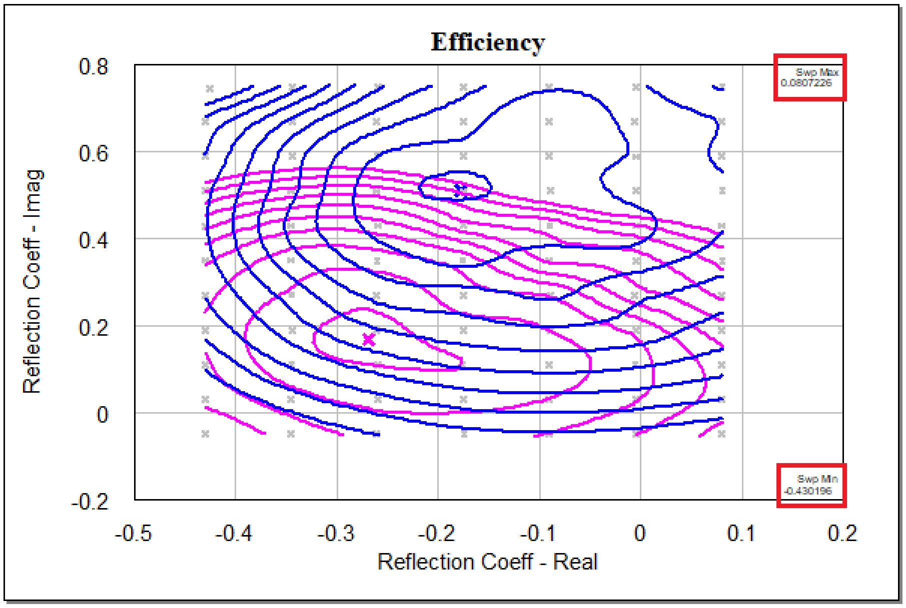
Rectangular graphs are x-y graphs that are used to display measurements that have real-valued results. Typically, the x-axis represents frequency or time, but can also be used to display any real-valued swept parameter. Measurements can be displayed on both the left and right y-axis. The following shows an example of a rectangular graph.



7.1.2.2. Rectangular - Real/Imag Graphs

Rectangular - Real/Imag graphs allow complex measurements (those that return values with real and imaginary components) which would normally be plotted on a Smith Chart or polar grid, to be plotted on a rectangular grid. Benefits of using a rectangular plot include:

- independent adjustment of the limits and step sizes on the vertical and horizontal axes
- ease of reading the real and imaginary values directly from the axis labels. This graph type is particularly useful for plotting load pull data.

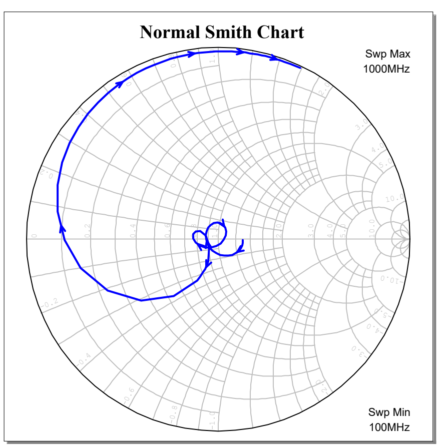


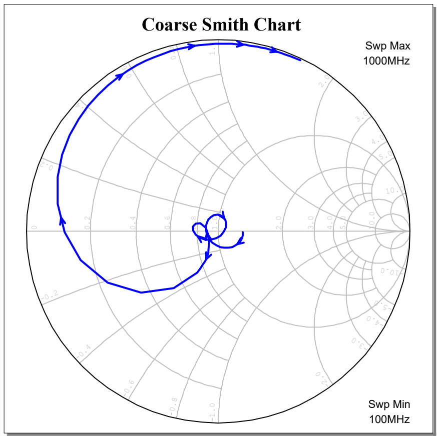
Notice that the minimum and maximum sweep values for the independent variable display at the right side of the grid, similar to their display on a Smith Chart.

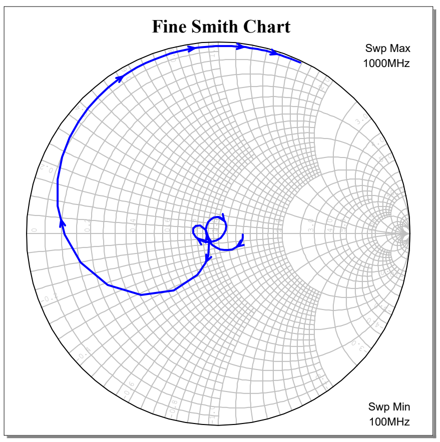
7.1.2.3. Smith Charts

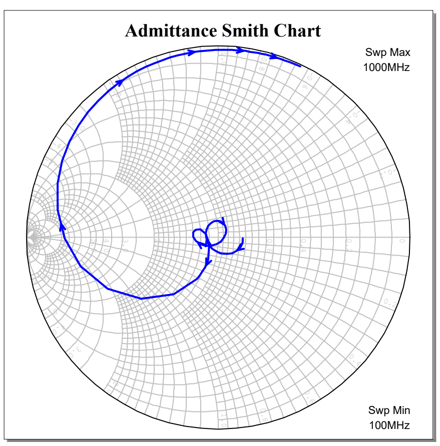
The Smith Chart is a graph that allows all passive impedances or admittances to be plotted in a reflection coefficient chart of unit radius.

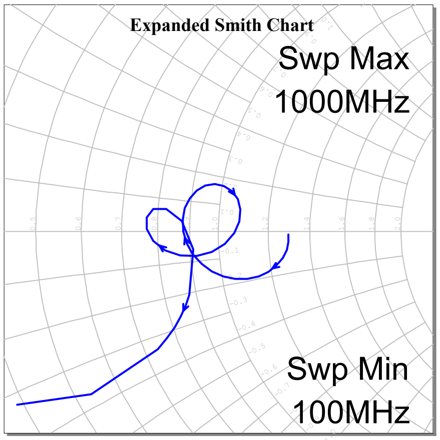
You can display a Smith Chart in several different formats. In addition to the standard Smith Chart with a unity radius, you can display an expanded Smith Chart and a compressed Smith Chart. A Smith Chart can be displayed as an impedance chart, an admittance chart, or both. The data cursor for the Smith Chart can display trace information as impedance, admittance, or as a reflection coefficient. The following figures show examples of different types of Smith Charts.

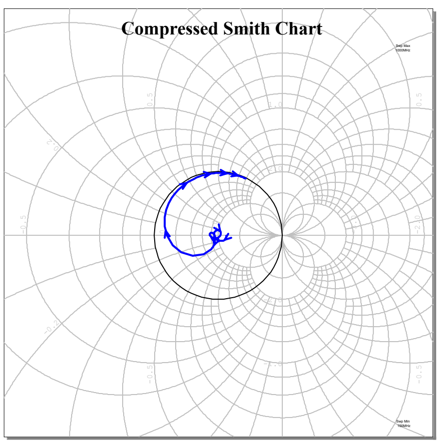



Coarse Smith Chart
Swp Max
1000MHz
Swp Min
100MHz


Fine Smith Chart
Swp Max
1000MHz
Swp Min
100MHz

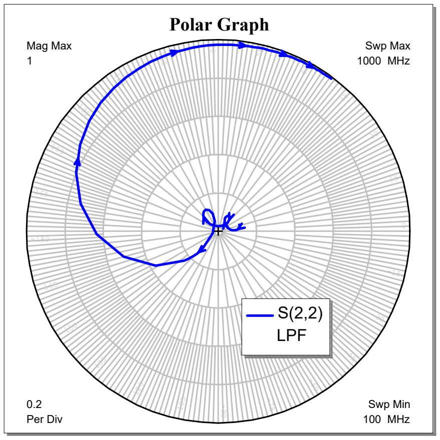

Admittance Smith Chart
Swp Max
1000MHz
Swp Min
100MHz





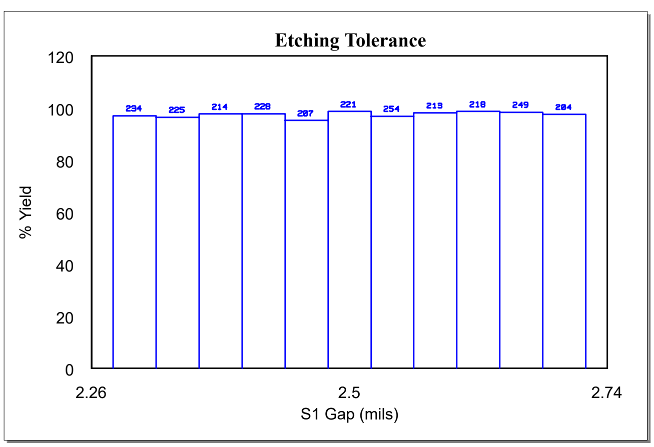
7.1.2.4. Polar Grids

A polar grid allows measurements that have complex results to be plotted on a graph that displays the magnitude and angle of the measurement. The following figure is an example of a polar grid.



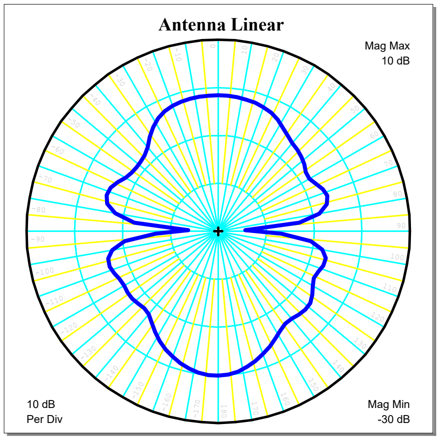
7.1.2.5. Histogram Graphs

Histograms are a form of bar chart normally used in yield analysis, as described in "Analyzing the Results". Yield histograms plot the yield percentages of a target performance parameter as a function of a variable process or device parameter. When used to plot a yield sensitivity measurement such as YSens, the values of the process or device parameter to be varied are shown on the x-axis of the histogram. The x-axis is divided into a set of bins, with each bin representing a range of values of the variable process or device parameter. Each bin is associated with a bar whose height (y value) is the percent yield of the target performance parameter for values of the variable process or device parameter within the range of its bin. A number also displays at the top of each bar to represent the total number of trials used to compute the yield for that bin.



7.1.2.6. Antenna Plots

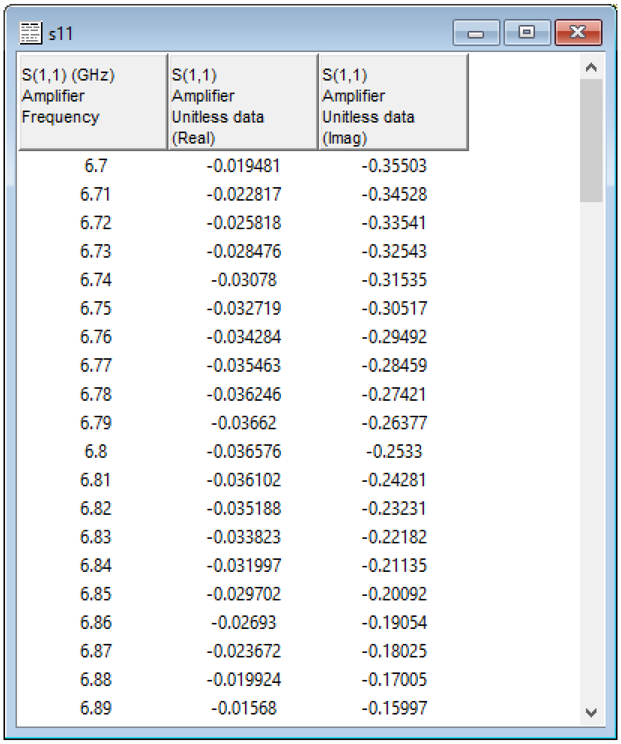
An antenna plot allows measurements that have real results to be plotted on a polar grid that displays the sweep dimension of the measurement as the angle and the data dimension as the magnitude. The following figure is an example of an antenna plot.



7.1.2.7. Tabular Graphs

Tabular graphs display measurements as columns of numbers. The first column of a given measurement is comparable to the x-axis on a rectangular graph, and can represent frequency, time, or some other swept parameter. The remaining columns for each measurement are used to display the measurement data. The header at the top of each column identifies the particular measurement or sweep parameter and its data format. You can right-click a column header for a context menu with additional measurement commands. When all measurements of the table have the same sweep values, the first column of the table is a common sweep column shared among all measurements. When the sweep values for some measurements differ from others, each measurement has a separate column for its sweep data. The **Hide x-axis column** command can be used to hide these separate x-axis columns.

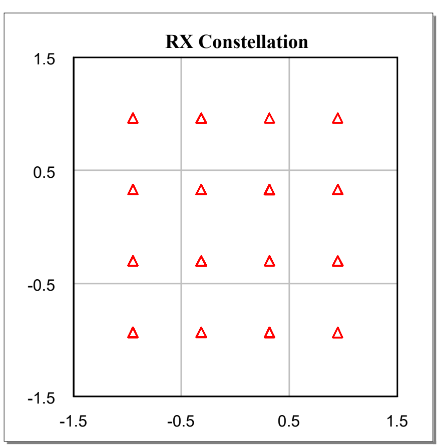
You can specify sweep precision (elements in the first column such as frequency, time, and voltage) separately from data precision (second and subsequent columns) in the Tabular Graph Options dialog box. The following figure is an example of a tabular output window.



S(1,1) (GHz) Amplifier Frequency	S(1,1) Amplifier Unitless data (Real)	S(1,1) Amplifier Unitless data (Imag)
6.7	-0.019481	-0.35503
6.71	-0.022817	-0.34528
6.72	-0.025818	-0.33541
6.73	-0.028476	-0.32543
6.74	-0.03078	-0.31535
6.75	-0.032719	-0.30517
6.76	-0.034284	-0.29492
6.77	-0.035463	-0.28459
6.78	-0.036246	-0.27421
6.79	-0.03662	-0.26377
6.8	-0.036576	-0.2533
6.81	-0.036102	-0.24281
6.82	-0.035188	-0.23231
6.83	-0.033823	-0.22182
6.84	-0.031997	-0.21135
6.85	-0.029702	-0.20092
6.86	-0.02693	-0.19054
6.87	-0.023672	-0.18025
6.88	-0.019924	-0.17005
6.89	-0.01568	-0.15997

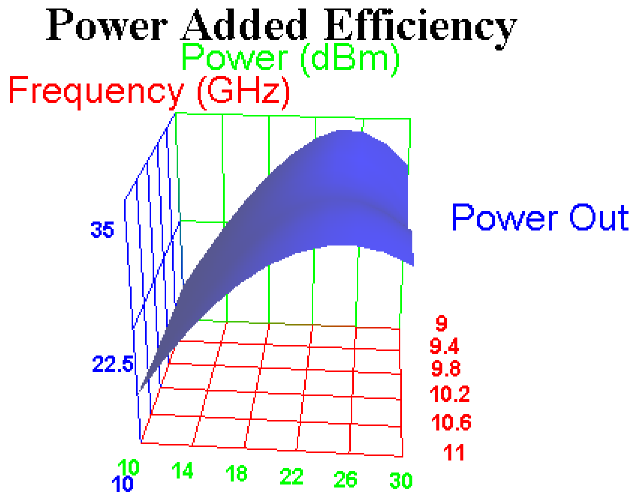
7.1.2.8. Constellation Graphs

Constellation graphs plot the real and imaginary parts of a complex-valued signal against each other, usually with time as an implied parameter. The horizontal axis shows the real part of the signal, while the vertical axis shows the imaginary part. In addition, symmetry constraints are applied to the scaling of the minimum and maximum x and y axis values. Specifically, the minimum and maximum values on each axis are negatives of each other, and the x and y axes have identical minimum and maximum values. By default, the line style of constellation graphs is set to "scatter" so the data displays as unconnected dots. Constellation plots are most useful when displaying measurements such as the IQ component of a baseband signal whose modulation scheme results in a predictable pattern of the real and imaginary parts of the signal.



7.1.2.9. 3D Graphs

3D graphs plot one or more real-valued measurements as a function of two parameters. Results are displayed as a surface, analogous to a function of the form z = f(x, y). The two swept parameters are associated with the x- and y-axes of the graph by using the Add/Modify Measurement dialog box. The value of the measurement at each x-y pair becomes the z-axis value of the graph.



7.1.2.10. Changing Graph Types

To change the type of an existing graph, right-click the graph in the Project Browser and choose **Change Type To** and choose an available graph type. A graph window for the new graph type opens in the workspace and the graph type icon changes to that of the new graph.

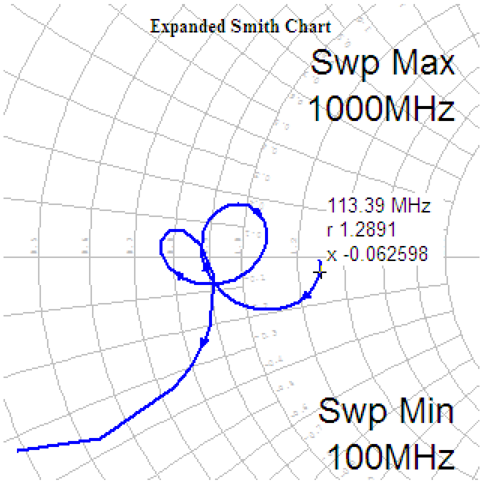
7.1.3. Reading Graph Values

To display a value on a graph you can click on a trace or you can add markers to graphs to permanently see values on traces.

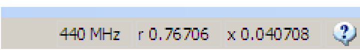
7.1.3.1. Cursor Display

The AWR Design Environment platform features a data cursor you can use to easily read numerical values for a particular point on a trace.

To use the data cursor, click near the trace in the plotting area of the graph. The cursor changes to a "**+**" and the closest data values display next to the cursor. As you slide the cursor along the measurement trace, it tracks the data points.



Additional information also displays in the Status bar at the bottom of the window.



You can use the data cursor with rectangular graphs, Smith Charts, polar grids, and antenna plots.

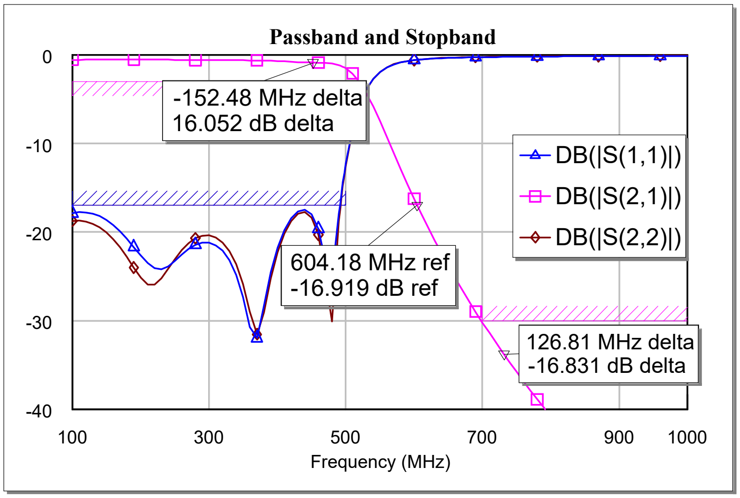
7.1.3.2. Adding Graph Markers

You can add graph markers to traces on graphs by choosing **Graph > Marker > Add Marker**, then clicking and dragging on a trace on the graph. The location of the marker displays while you drag the mouse, and the marker is added when you release the mouse button. You can change the position in which the marker is placed on a specific trace by clicking and dragging on the point where the marker connects to the trace. You can also delete a selected marker by pressing the **Delete** key. The size of the marker display box is controlled by clicking on it and dragging one of its resize handles to a new position. The font size of the marker changes as the marker is resized. See "Graph Options Dialog Box: Markers Tab" for more information.

To set graph data marker search mechanisms, select the marker display box, right-click and then choose the appropriate option:

- **Marker->max**: Moves the marker to the maximum value of the plotted function (the graph rescales automatically if maximum is not visible). This feature does not apply to Smith Charts or polar grids.
- **Marker->min**: Moves the marker to the minimum value of the plotted function (the graph rescales automatically if minimum is not visible). This feature does not apply to Smith Charts or polar grids.
- **Marker Search**: Displays the Marker Search dialog box with options for specifying a specific x or y value to search for, the search direction (left or right), and the search mode (Absolute or Delta). On Smith Charts and polar grids, the Search for Sweep Value dialog box displays with options for specifying a sweep value and the search mode (Absolute or Delta). To specify the minimum and maximum sweep value limits to control the range of frequencies over which Smith Chart data is swept, right-click in a Smith Chart window and choose **Options**. In the Smith Chart Options dialog box on the **Grid** tab, clear the **Default** check box (the project option values) and enter the **Min** and **Max** sweep value limits.
- **Reference Marker**: Makes the selected marker the reference from which other markers are valued.

The following graph shows example markers with text.

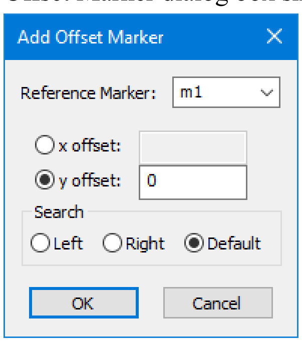


Auto-search Markers

You can add an auto-search marker on a selected trace by right-clicking in a rectangular graph and choosing **Add Auto-Search Marker** or by clicking the **Add Auto-Search Marker** button on the Graphs toolbar. These markers automatically search for a user-specified feature on the trace and shift to stay on the feature as the curve updates during changes such as tuning and optimization. See "Add/Edit Auto-Search Marker Dialog Box" for the features you can specify. Once created, you can edit an auto-search marker by right-clicking in the graph and choosing **Edit Auto-Search Options**.

Offset Markers

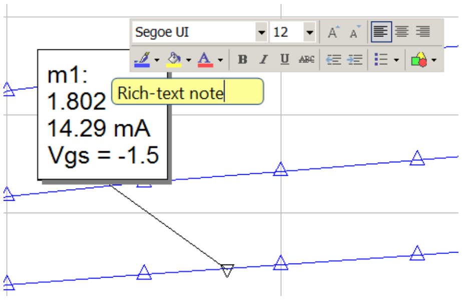
You can add an offset marker on a selected trace by right-clicking in a rectangular graph and choosing **Add Offset Marker** or by clicking the **Add Offset Marker** button on the Graphs toolbar. These markers automatically maintain a specified offset from another marker on the trace. The distance and the marker from which they are offset is specified in the Add Offset Marker dialog box shown in the following figure.



When you choose **Add Offset Marker** from a marker's context menu, the name of the marker displays in **Reference Marker**. After placing an offset marker, you can edit its properties by choosing **Edit Marker Offset** from the context menu. An offset marker can reference another offset marker or an auto-search marker.

Marker Notes

You can choose **Add Note** from the context menu of a marker or its label to attach a rich-text note that provides a customized description of the marker. You can customize the note font attributes. The note maintains its position relative to the marker location as the trace data updates.

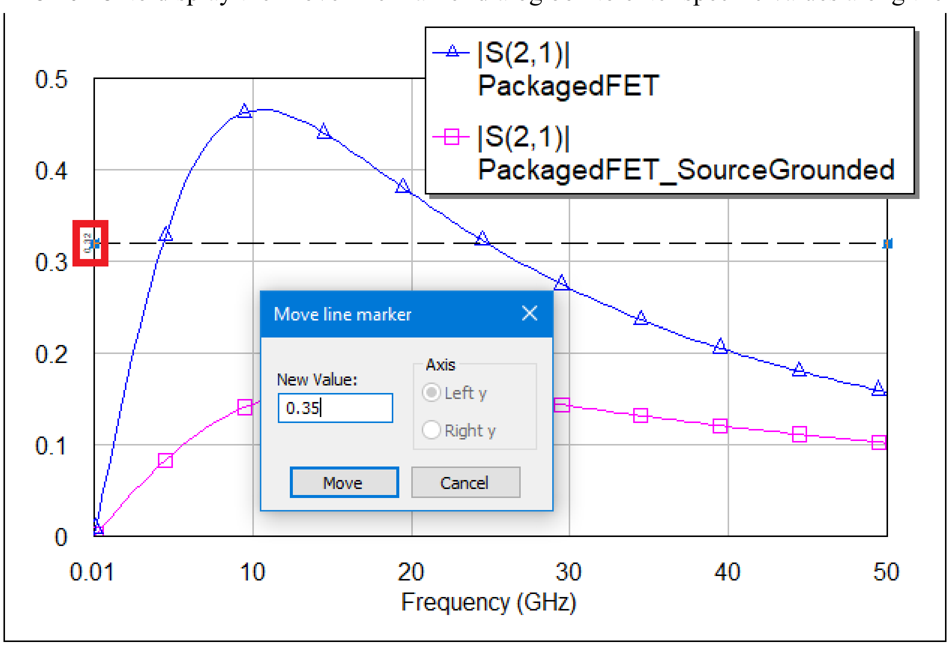


Marker Names in Labels

By default, marker names display in the labels. It is helpful to know the marker names because they are referenced in the specifications for offset markers and may be used in equations and sweep selectors. You can hide marker names by clearing the **Names in labels** option on the Rectangular Plot Options dialog box **Markers** tab.

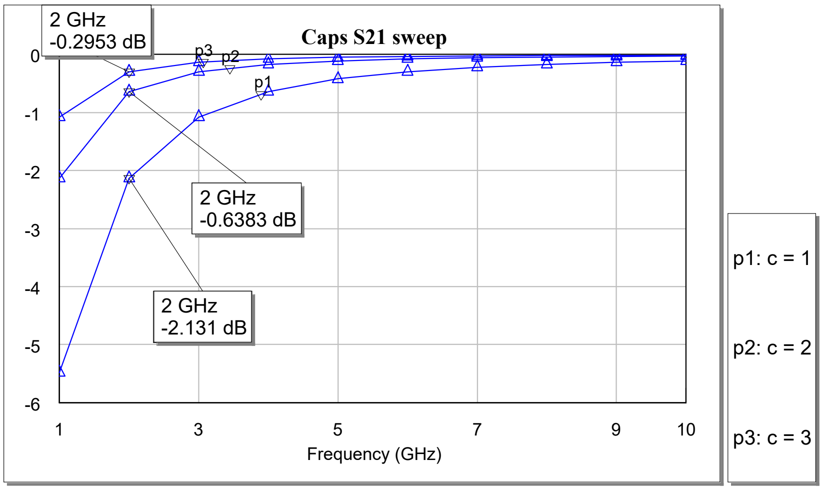
7.1.3.3. Adding Line Markers

You can measure individual traces by adding horizontal and/or vertical line markers. Right-click on a graph and choose **Add Horizontal Line Marker** or **Add Vertical Line Marker**, and then click at the horizontal or vertical point at which you want to add a reference line. Horizontal line markers display the y-axis value, and vertical markers display the x-axis value at the graph axis. Click anywhere on the marker line to view the value at that position, or click and drag along the marker line to view continuous values. You can move the entire marker line along the same axis by clicking and dragging it, or by pressing the **Shift** - Up/Down arrow keys to move a horizontal line marker up or down on a grid defined by one tenth of the y-axis division step size. Right-click on a line marker in a histogram or rectangular type graph and choose **Move Line Marker** to display the Move line marker dialog box to enter specific values along the selected axis.



7.1.3.4. Adding Swept Parameter Markers

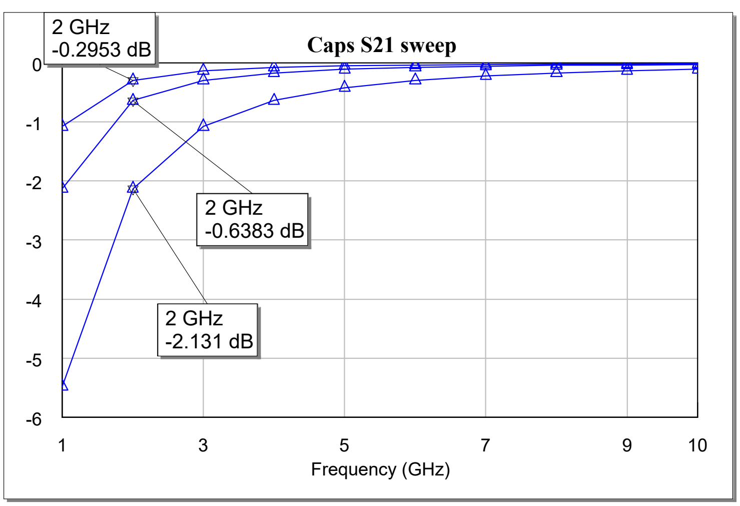
When you have swept parameters (such as IV curves or when using a SWPVAR block on a schematic), there can be many traces for one measurement. In this case, the graph automatically adds swept parameter markers so you know which sweep value is used to create which trace. This differs from graph markers; the graph displays the sweep value for a trace instead of the x,y values of the graph. You can move parameter markers along a trace. The following example shows both swept parameter markers and graph markers with default settings.



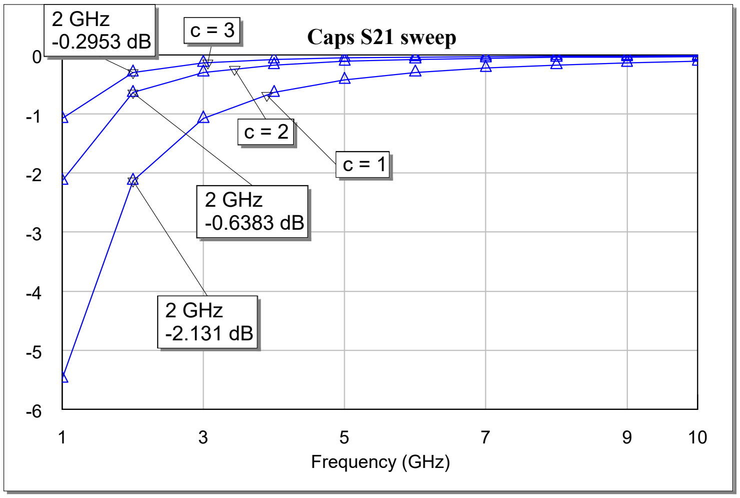
7.1.3.5. Modifying Marker Display

You can change many different graph and parameter marker characteristics, mainly on the graph Options dialog box **Markers** tab. See "Graph Options Dialog Box: Markers Tab" for details.

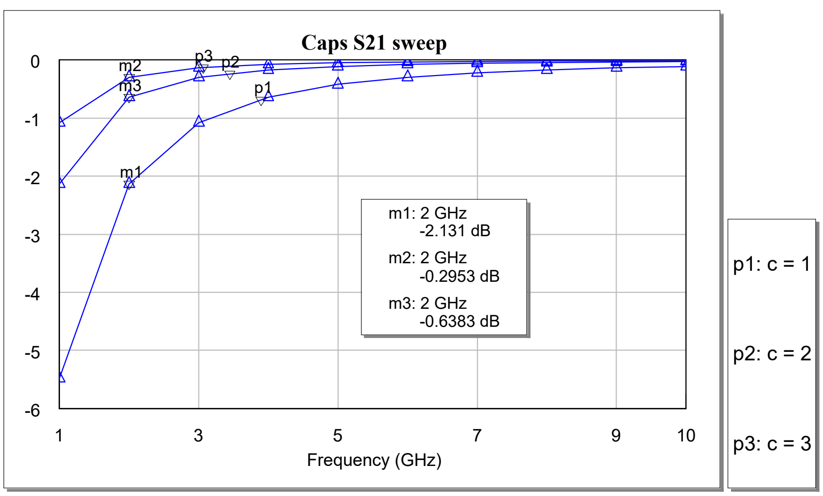
The **Param markers enabled** check box specifies if the parameter markers display. The following example shows this option turned off (not selected).



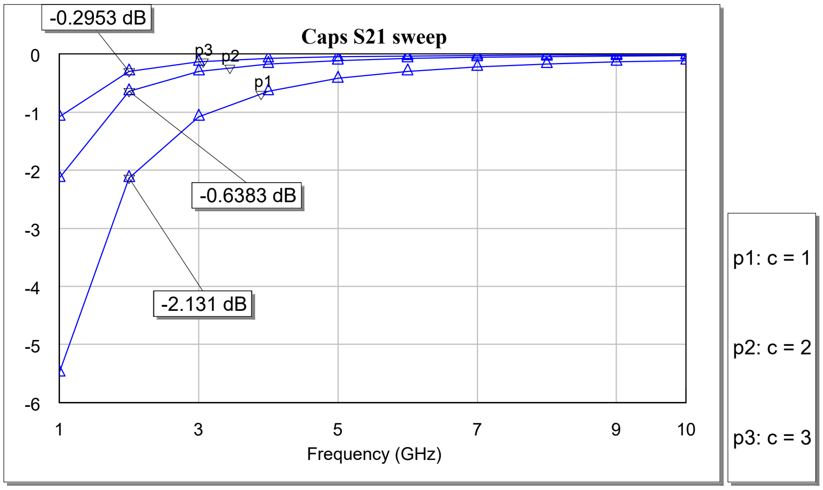
The **Param markers in legend** check box controls if the parameter markers display in a legend or on the graph with lines attached for each value. The following example shows this setting turned off (not selected).



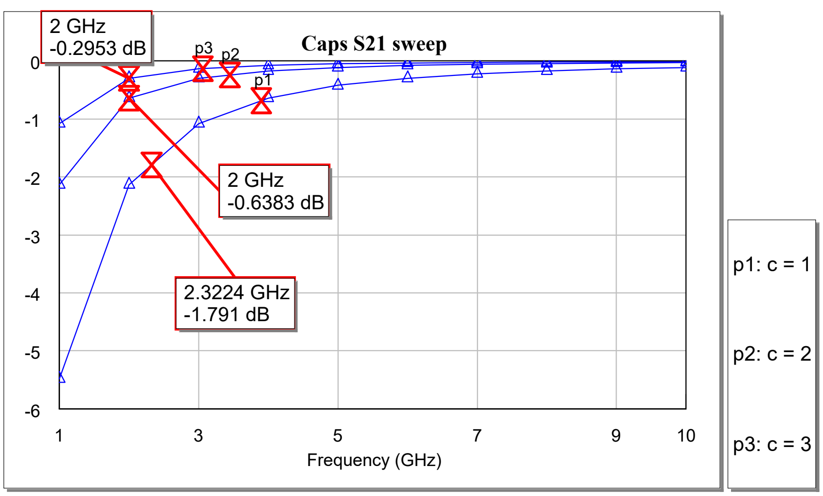
The **Data markers in legend** check box controls if the data graph markers display in a legend or on the graph with lines attached for each value. The following example shows this setting turned on (selected).



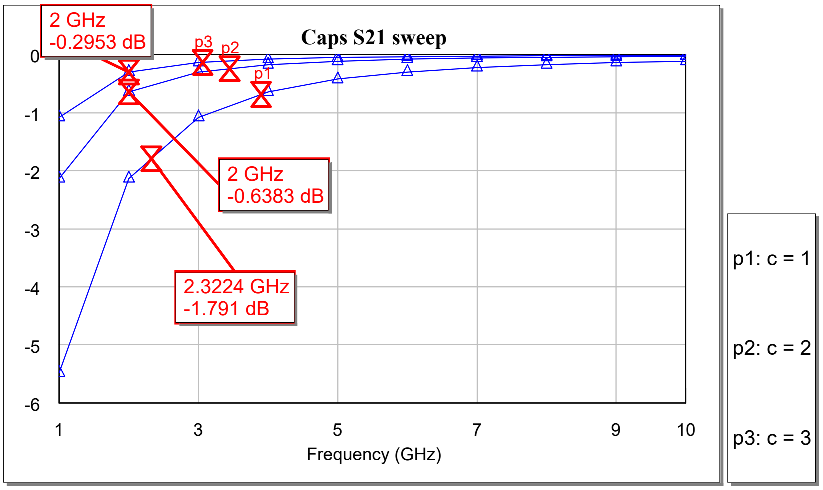
The **Show sweep val** check box controls if the data graph markers display the x-axis value. The following example shows this setting turned off (not selected).



The **Color**, **Weight**, **Symbol** and **Size** options control the appearance of the lines and markers on the graph. The following example shows changes in these settings from the previous figure.

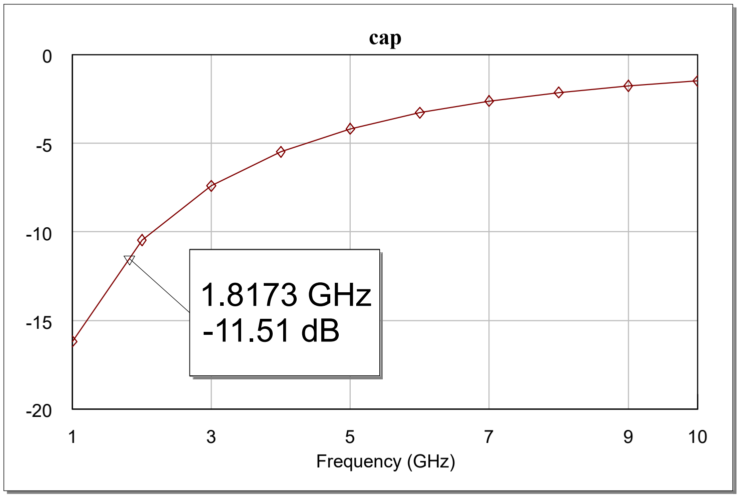


Note that these settings do not change the marker text font. This is done on the graph Options dialog box **Fonts** tab. See "Graph Options Dialog Box: Fonts Tab" for details. In this dialog box, click the **Markers** button to display a Font dialog box with options for changing the characteristics of the marker fonts. The following example shows the marker text font changed to red.



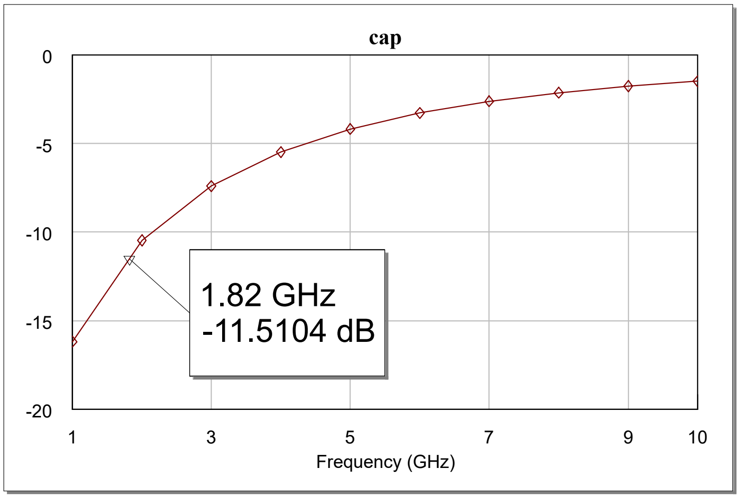
7.1.3.6. Modifying Number of Digits in Cursor and Marker Display

The number of significant digits displayed for data cursors and markers is controlled on the graph Options dialog box **Numeric** tab. See "Graph Options Dialog Box: Numeric Tab" for more details. The marker in the following example shows the default settings for the precision of the values.



There are separate settings for the sweep values in the **Marker/Cursor Sweep Value Format** section and the data values in the **Marker/Cursor Data Value Format** section. The sweep values are the values used for the graph's x-axis (for rectangular plots), or whatever is used for the x-axis in the measurement setup. In the previous example, frequency is the sweep value and is the top value shown in the marker. The data values are typically the y-axis values or the values created from the simulators. In the previous example, the magnitude of s11 in dB is the data value and it is the bottom value shown in the marker.

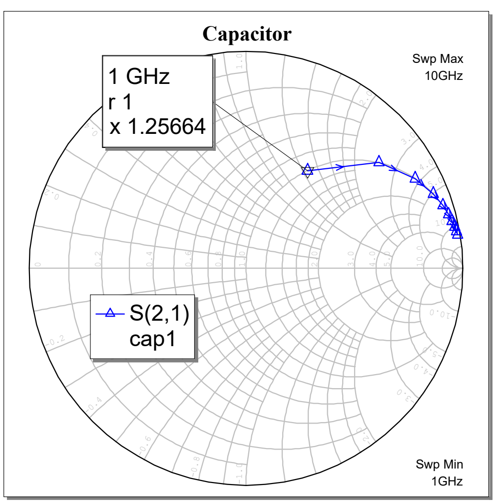
The following example shows the same marker with sweep significant digits limited to three, and the data showing four significant digits to the right of the decimal place.



Tabular graphs have a separate Options dialog box, but they also include control for the display precision for the sweep and the data. See “Tabular Graph Options Dialog Box” for details.

7.1.3.7. Modifying Cursor and Marker Display for Complex Data

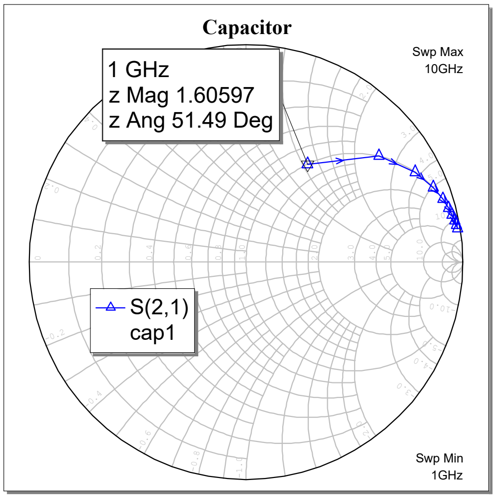
When using cursor display or markers on a Smith Chart, there are many different ways to display data. For example, the following Smith Chart shows a marker at 1 GHz.



This cursor display shows the normalized impedance in terms of real (r) and imaginary (x).

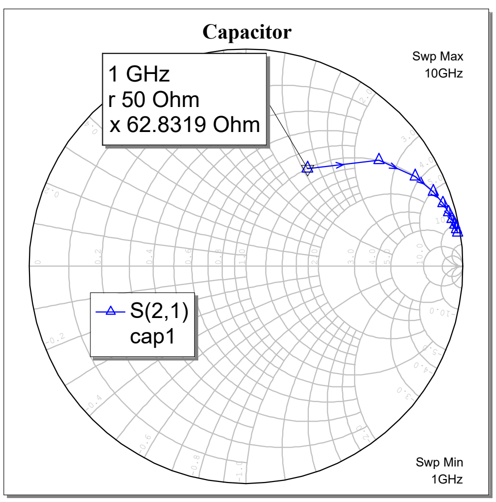
You can change these options on the graph Options dialog box **Markers** tab. See “Graph Options Dialog Box: Markers Tab” for more information.

The **Display Format** section controls how to display complex data. The following example shows this value set to **Magnitude/Angle**.

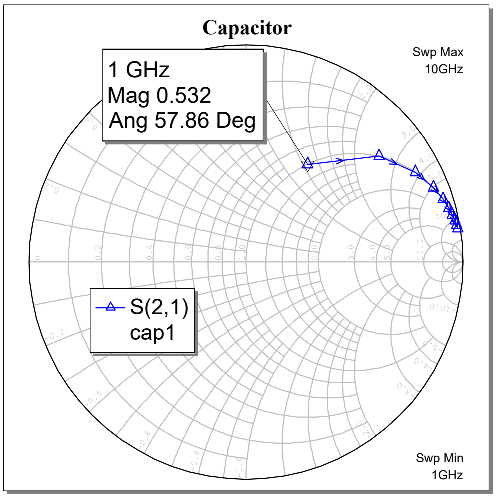


These settings also apply to polar grids as well.

The **Z or Y display** section controls how to display impedance or admittance values. You can denormalize the values. The following example shows the **Denormalized to** option selected with a value of 50 ohms used.



The **Display Type** section specifies the type of values to display from a Smith Chart: impedance, admittance, or reflection coefficient. The following example shows **Display Type** as **Reflection Coefficient** and **Display Format** as **Magnitude/Angle**.



7.1.4. Modifying the Graph Display

Graph display in the AWR Design Environment platform is completely configurable. You can change settings such as the colors, line styles, fonts, labels, markers, data cursor settings, zoom level, and chart details.

To modify a graph display:

1. Right-click in a graph window and choose **Options**, or double-click the edge of the grid on a rectangular graph, polar grid, or Smith Chart.

   The graph Options dialog box displays. This dialog box has a number of tabs you can click to modify different settings. See “Graph Options Dialog Box: Format Tab” for information about the **Format** tab on this dialog box.

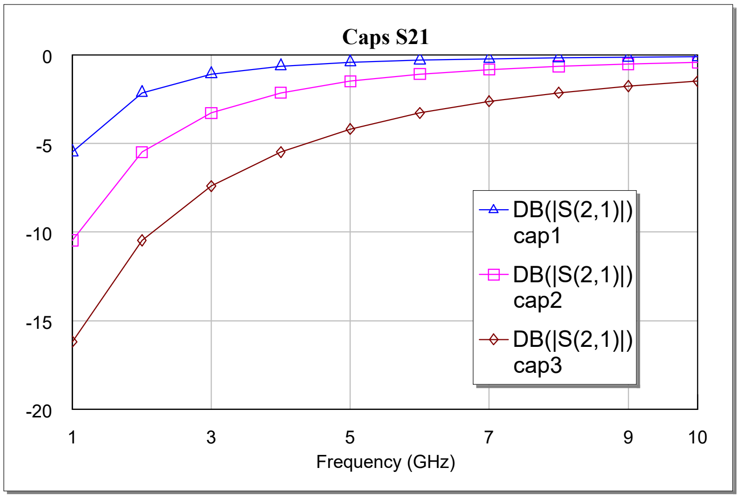
2. Make any desired changes on the associated tabs, and click **OK**.

7.1.4.1. Graph Traces

There is one set of trace properties for each measurement added to a graph. When you use swept variables, you might have many sweep points on a graph, but these are still considered one trace, so all the data from a swept variable must have the same settings. There are several different settings you can apply to traces individually, including:

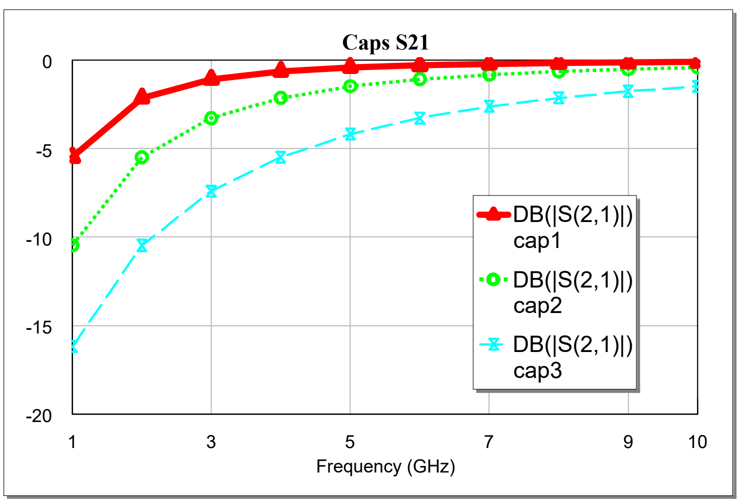
- style (line thickness, color, etc.)
- symbol attributes
- trace type
- which graph axis to use
- how the measurement data displays in the legend.

The following rectangular graph shows three measurements with the default settings for rectangular graphs.



Trace Style

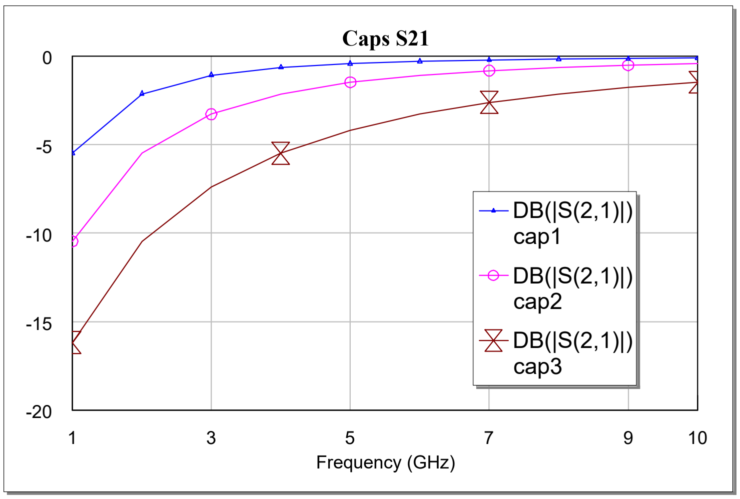
Trace style options include color, symbol, line, and weight. See the "Graph Options Dialog Box: Traces Tab" for more information. The **Measurement** area displays the measurement associated with the trace selected in the **Style** section. The following example shows the previous graph with different trace style settings for the three measurements.



Load Pull Measurements: Load pull contours use the **Stepped color** options in a slightly different way than other measurements. For load pull contours, the actual magnitude of the contour values are used to generate the different variations of color. For example, if you have a load pull measurement over swept frequency that plots two contours for a single measurement (one for each frequency), then the colors shown for each contour trace reflect the magnitude of the values instead of just the trace index like other measurements. This allows the trace color to be used to indicate relative magnitude, even when multiple contours are drawn with a single measurement.

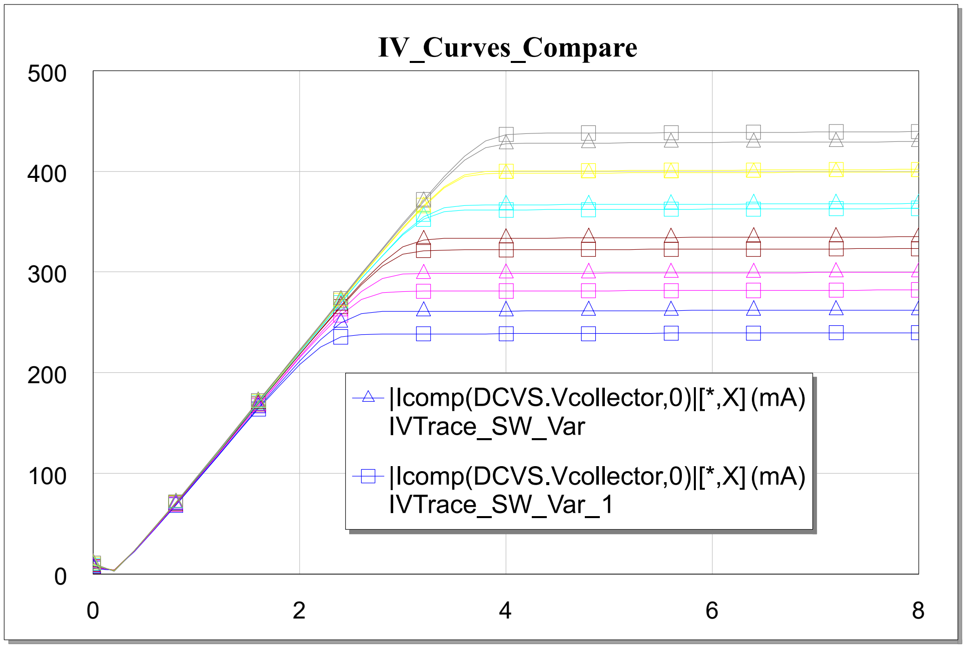
Trace Symbol

Trace symbol options control the symbol interval and size. See the "Graph Options Dialog Box: Traces Tab" for more information. The visual symbol is set in the **Style** section. The **Auto interval** option attempts to keep 11 symbols on a trace, regardless of the number of points on the trace. Sometimes it is difficult to know which is a simulation point and which is a point extrapolated between points. If you set the symbol interval to 1, you know exactly which points were simulated. The following example shows the same graph with different settings for the traces of the three measurements.



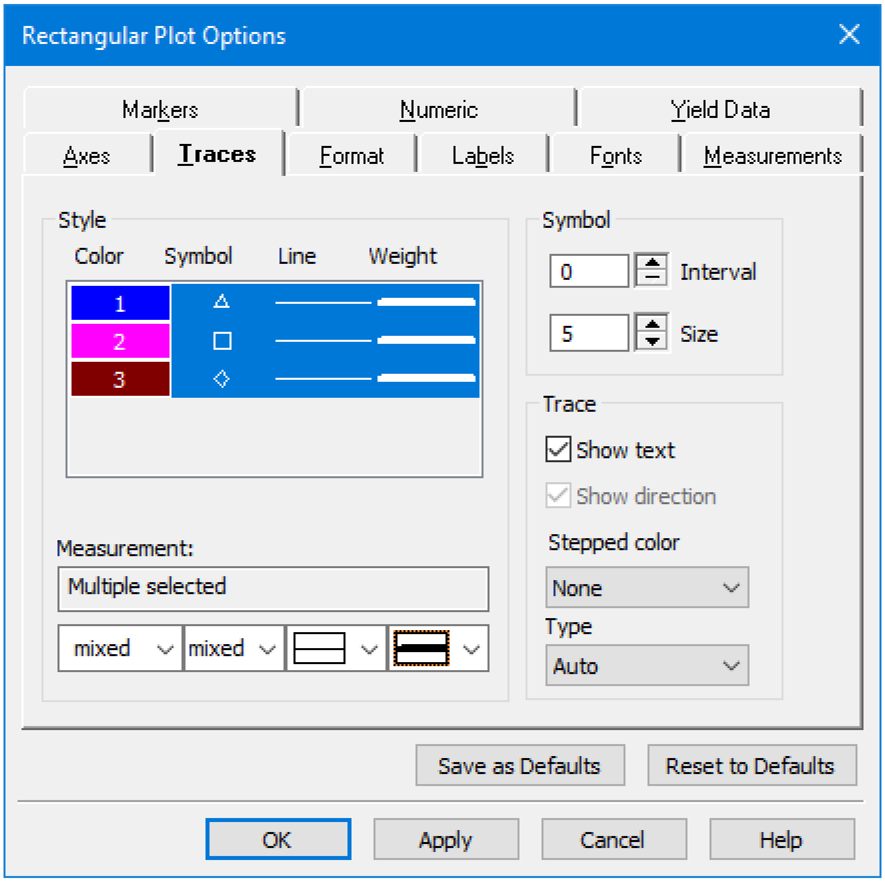
Step Color on Traces

The Step Color feature specifies whether trace colors change for each trace in a measurement. This feature is set in the **Trace** section of the graph Options dialog box by selecting the **Step Color** check box. See the "Graph Options Dialog Box: Traces Tab" for more information. If you select the **Step Color** check box, the first trace matches the color specified, and subsequent colors in the list are used for subsequent traces. To compare swept parameter measurements and have matching trace colors for corresponding parameter steps, you must choose the same color for each measurement. The following figure shows a graph using Step Color.



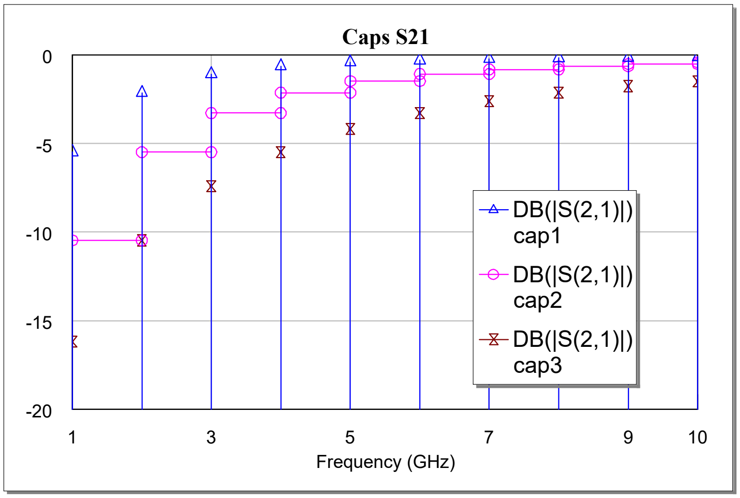
Selecting Multiple Traces

You can select multiple traces in a graph Options dialog box. See the "Graph Options Dialog Box: Traces Tab" for more information. To select more than one trace in the **Style** section of the dialog box, **Ctrl**-click to add single traces or **Shift**-click to select a consecutive range of traces. Options that you set apply to all selected traces. If a property differs between selected traces, the word "mixed" displays in the Color and/or Symbol blocks in the **Measurement** section of the dialog box instead of a color or symbol, as shown in the following figure.



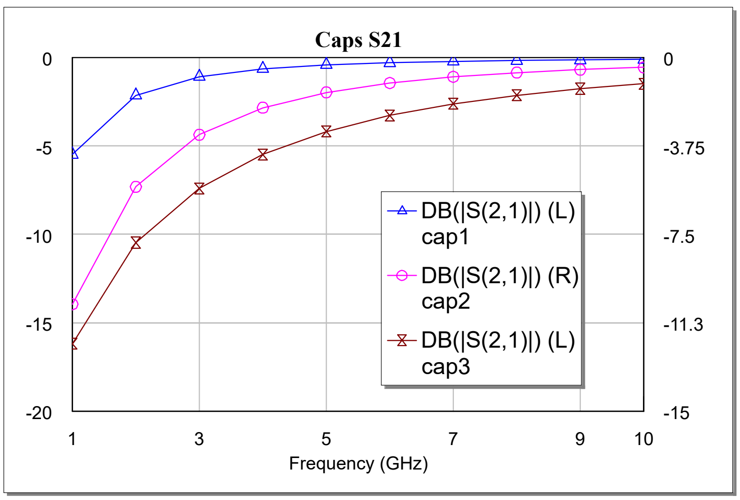
Trace Type

The trace **Type** option specifies how each simulation point is connected on the graph. See the "Graph Options Dialog Box: Traces Tab" for more information. Each measurement type picks an appropriate trace type, so typically the default setting is acceptable, although you can plot traces with any style you want. The following example shows the previous graph with different settings for traces for the three measurements.



Measurement Axis

By default, each measurement uses the left y-axis to display y-axis values. There are several options available. The first is to use the right y-axis for a given measurement. See the "Graph Options Dialog Box: Measurements Tab" for more information. The following example shows the same graph with the cap2 measurement using the right axis. Notice that in the legend there is an (R) or (L) after the measurement type to indicate which axis is used.

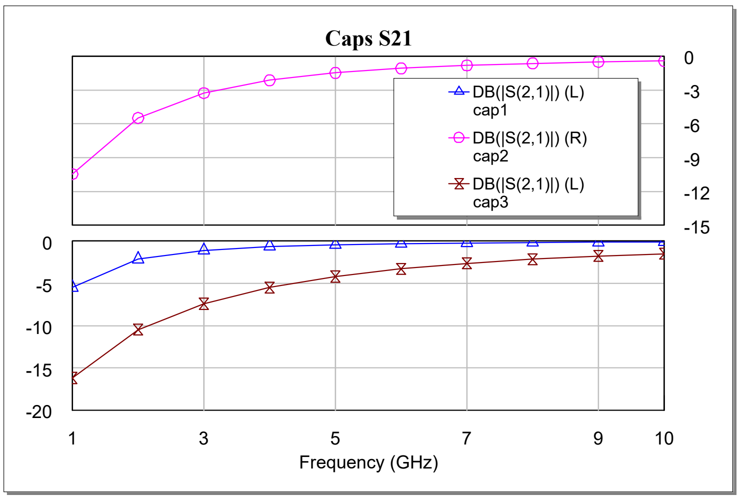


On all graph types except for 3D plots and tabular graphs you can edit property values by double-clicking directly on the number. On rectangular graphs you can edit the minimum and maximum values on the horizontal and vertical axes. When one of these values changes, the associated **Auto Limits** option is automatically turned off. You can also edit the second value on the axes to adjust the step size (for an axis with a linear scale) or Log Divs setting (for an axis with a logarithmic scale). On a Smith Chart or polar grid you can edit the sweep min and sweep max values. Values for the maximum magnitude value and step size per division are also editable on a polar grid, and on antenna plots you can edit the minimum and maximum magnitude values.

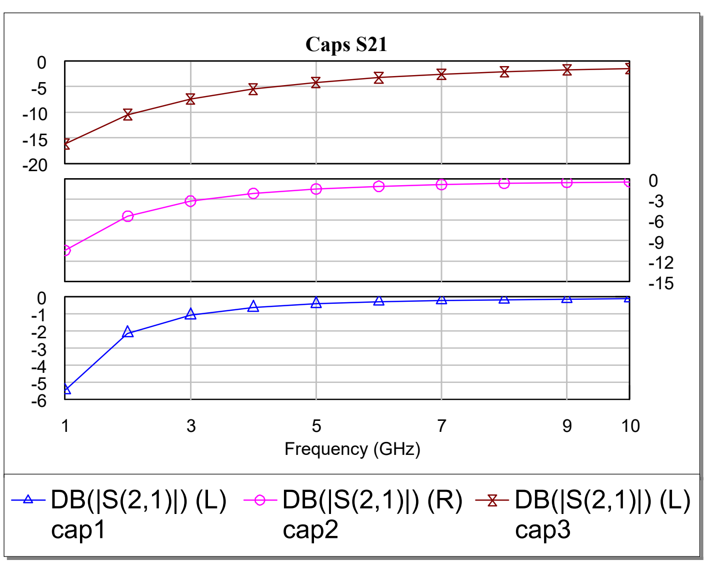
You can also add an axis to a graph to give you multiple graphs in one graph view sharing the same x-axis. See the "Graph Options Dialog Box: Axes Tab" for more information on defining a new axis. To add an axis, in the graph Options dialog box **Axes** tab, select the **Left 1** or **Right 1** axis in the **Choose axis** section and then click **Add axis**. You can set individual limits and divisions for each axis.

For SPICE time waveforms with the same voltage range, you can stack multiple/split graphs for easier viewing.

On the graph Options dialog box **Measurements** tab you can individually select on which axis each measurement displays. See the "Graph Options Dialog Box: Measurements Tab" for more information. The following example shows the same graph with the cap2 measurement using the second right axis.

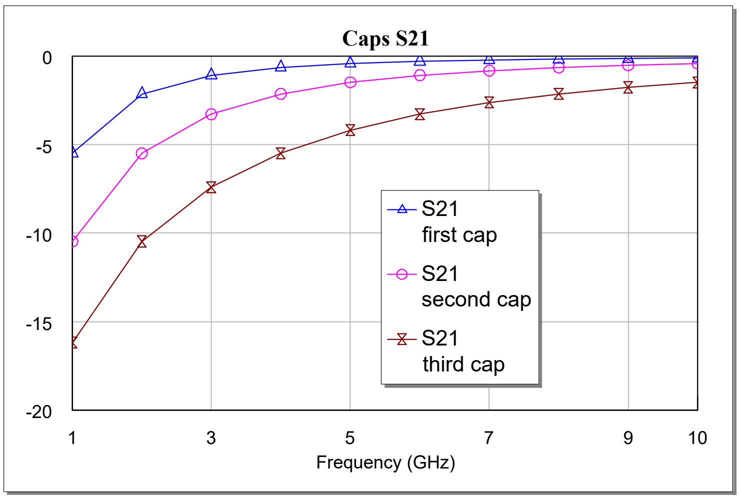


The **Measurements** tab includes an **Auto Stack** button that generates one new axis for each measurement on the graph, and moves each measurement to its own axis. The following example shows the same graph after clicking the **Auto Stack** button.



Measurement Legend Display

By default, the legend for each measurement shows the measurement string over the data source name. For each measurement, you can change this text. The **Legend Data Name** settings control how the data source displays, and the **Legend Meas Name** settings control how the measurement displays. See the "Graph Options Dialog Box: Measurements Tab" for more information. Note that there are legend options that control if the data source and/or measurement display in the legend. The following example shows the same graph with its legend updated to use alternate text.



7.1.4.2. Additional Measurement Options

You can access the following additional measurement options by right-clicking a measurement in the graph legend. Equivalent commands available by right-clicking a measurement in the Project Browser are shown in parentheses.

- **Modify Measurement** - displays the Modify Measurement dialog box for editing measurement properties. (**Properties**)
- **Toggle Enable Measurement** - enables or disables the measurement in the graph. (**Toggle Enable**)
- **Delete Measurement** - removes the measurement from the graph. (**Delete**)
- **Duplicate Measurement** - displays the Modify Measurement dialog box to allow you to add a new measurement. (**Duplicate**)
- **Simulate Measurement** - starts a simulation and displays the results on the graph. (**Simulate for measurement**)
- **View Source Document** - opens the data source document of the measurement. (**View Source Document**)
- **Add Optimization Goal** - displays the New Optimization Goal dialog box to allow you to add an optimization goal. (**Add Optimization Goal**)
- **Add Yield Goal** - displays the New Yield Goal dialog box to allow you to add a yield goal. (**Add Yield Goal**)
- **Format Measurement** - displays the graph Options dialog box **Measurements** tab with the measurement selected.
- **Modify Trace Properties** - displays the graph Options dialog box **Traces** tab with the measurement's trace selected.

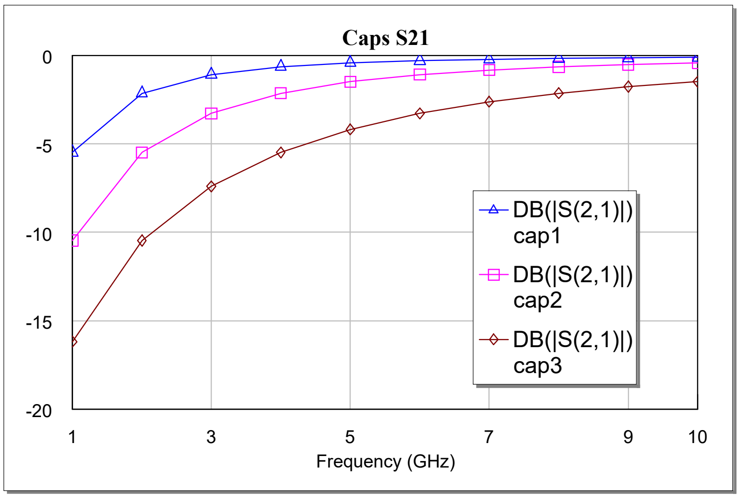
7.1.4.3. Modifying the Graph Legend

There is one set of graph options that control how the graph legend displays, including:

- what is displayed in the legend

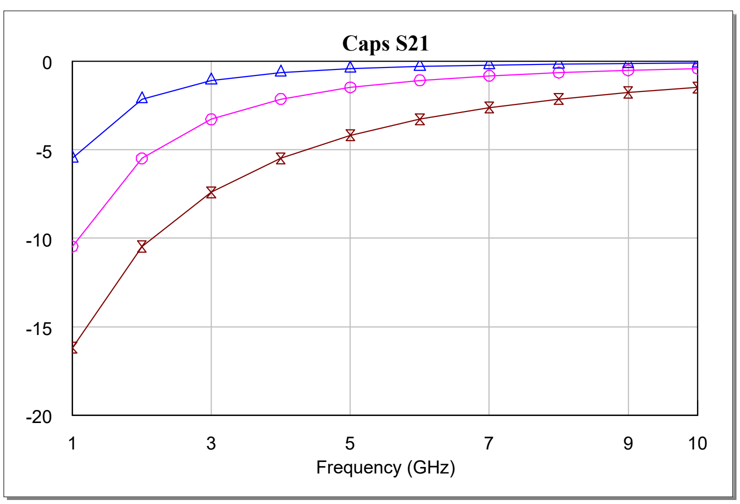
- size and location of the legend

The following rectangular graph shows three measurements with the default settings for rectangular graphs.

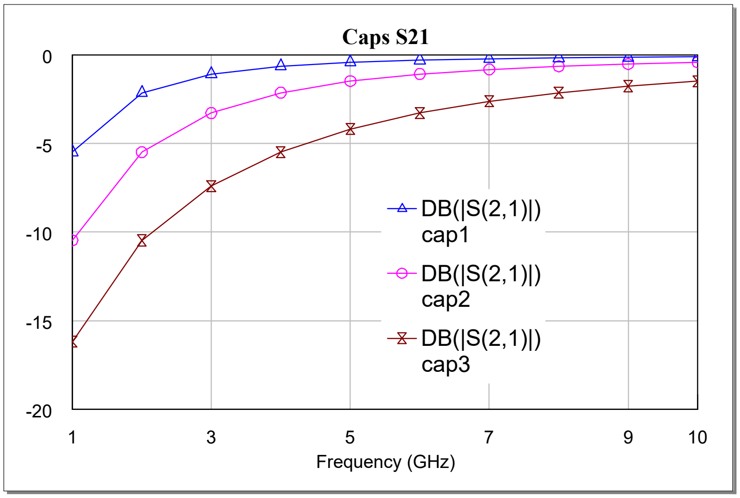


Legend Display

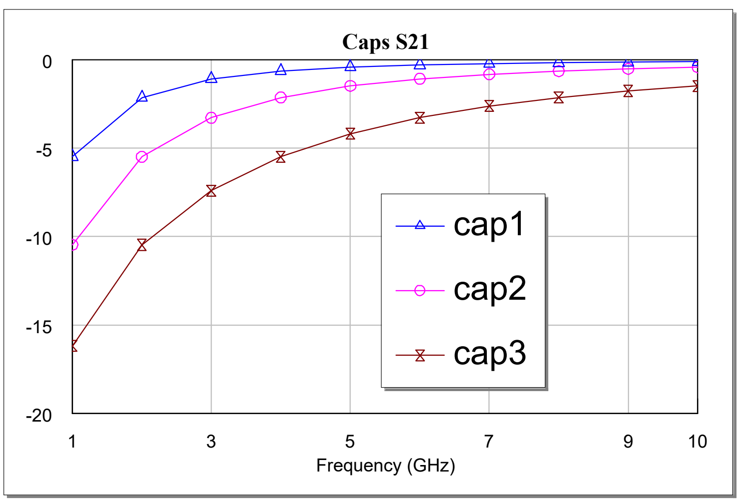
The **Legend** check box in the **Visible** section of "Graph Options Dialog Box: Format Tab" determines if the legend displays. The following example shows the graph with no legend (not selected).



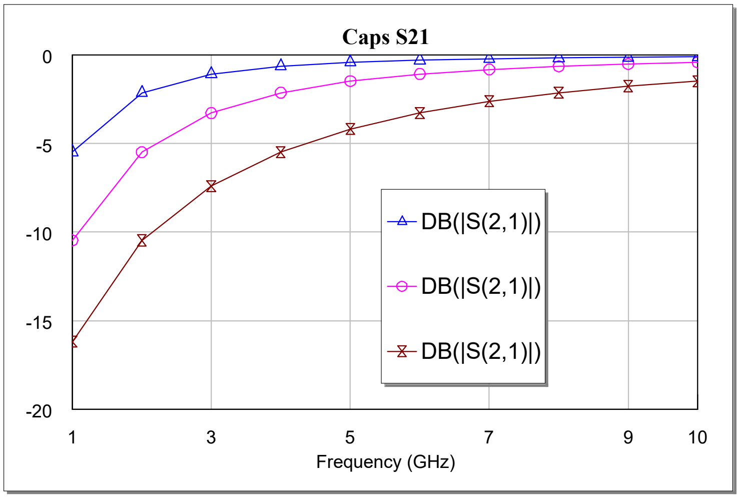
The **Legend border** check box in the **Visible** section of "Graph Options Dialog Box: Format Tab" determines if the legend border displays. The following example shows the graph with no legend border (not selected).



The **Data name** option in the **Legend entries** section of "Graph Options Dialog Box: Labels Tab" only displays the data source in the legend. The following example shows the graph with only the data source in the legend.



The **Measurement name** option in the **Legend entries** section of "Graph Options Dialog Box: Labels Tab" only displays the measurement name in the legend. The following example shows the graph with only the measurement name in the legend.



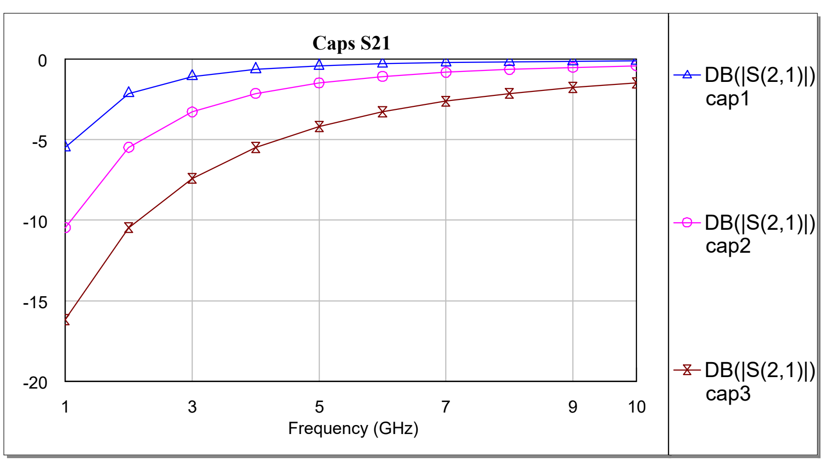
Legend Location and Size

You can modify a graph legend either using preset resize options, or manually.

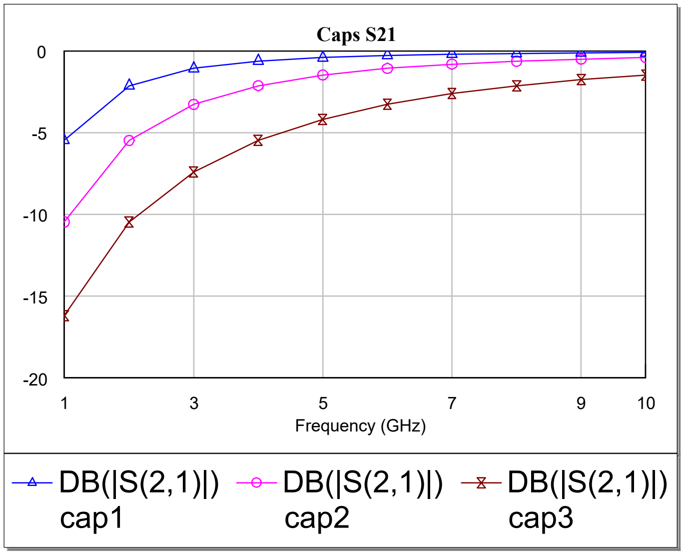
To automatically resize a graph legend so that text fits inside the legend box frame, or so the legend box frame fits the text, select the appropriate option under **Legend Frame** on the **Labels** tab of the graph Options dialog box. See "Graph Default Options Dialog Box: Labels Tab" for more information.

To manually modify a graph legend, click on it to display the drag handles on its border. To move the legend, just drag it to a new position. To change the size and aspect ratio, click a drag handle and drag it to a new position. An outline showing the new size of the legend displays as the mouse moves.

The legend changes how multiple measurements are listed based on the aspect ratio of the legend size. The following example shows a legend along the right side of the graph with the legend entries in a column.



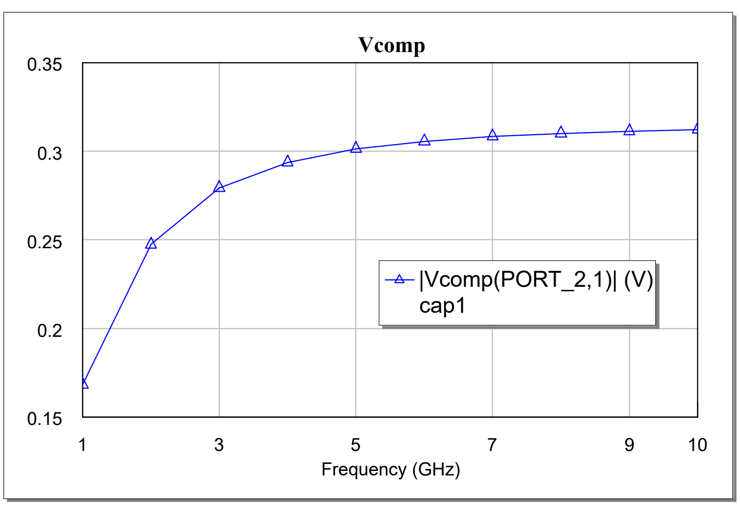
The following example shows a legend along the bottom of the graph with the legend entries in a row.



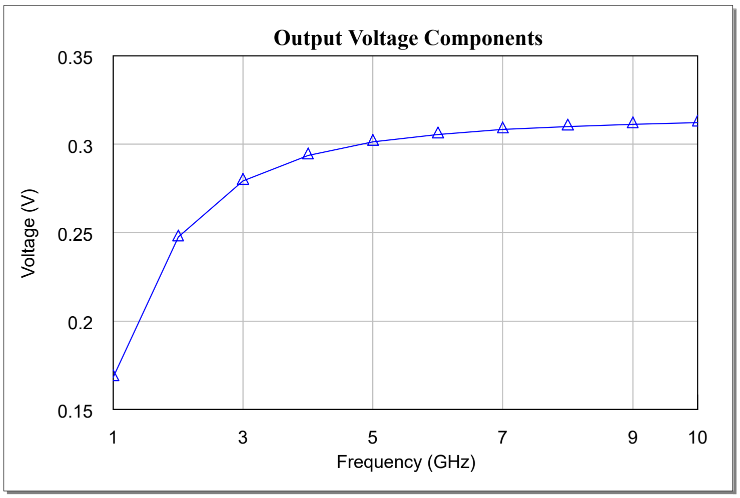
You can use alignment tools to make your legends the same width or height as the main graph window, and to align them with the graph window. The previous example graphs used alignment. To use the align commands, select everything in the graph by pressing **Ctrl+A**, then choose **Draw > Align Shapes** or **Draw > Make Same Size**.

7.1.4.4. Modifying Graph Labels

By default, the title of a graph is the name of the graph in the project. The x-axis label is determined by the x-axis of the graph, and there are no default y-axis labels. For example, see the following graph that plots the fundamental harmonic of the output voltage of a circuit with frequency on the x-axis. This graph is using all the default settings. Note that the units for the y-axis are shown in parentheses (V) in the legend.



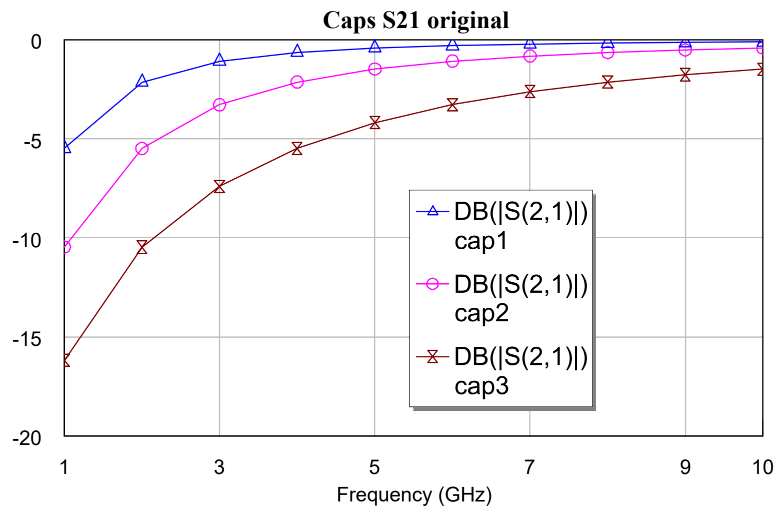
You can easily change these settings. See “Graph Options Dialog Box: Labels Tab” for more information. You can override the default graph title and type in the left or right y-axis labels. For the y-axis, the units cannot be generically determined (for instance, you might have many different measurements on one y-axis), so you need to type them in if you want them on the label. See the following example of the original graph with the title and labels changed and the legend turned off (not selected).



7.1.4.5. Modifying the Graph Border/Size

You can manually change the size of a graph display. To manually modify the display size of a rectangular graph, polar grid, or Smith Chart, click the border of the graph to display drag handles. To modify the size and aspect ratio of a graph border, click a drag handle and drag it to a new position/size. An outline showing the new size of the graph displays as the mouse moves.

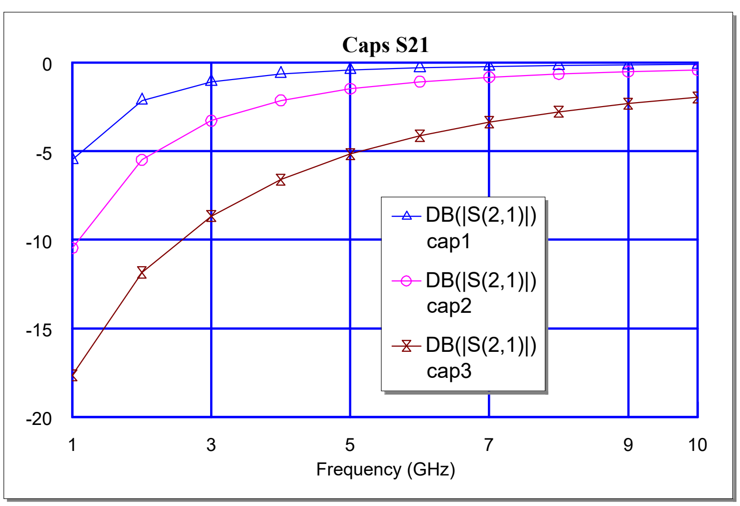
You can also turn off the grey background color of a graph. The **Border** check box in the **Visible** section of “Graph Options Dialog Box: Format Tab” determines if the graph background displays. The following example shows the graph border turned off (not selected). Notice that the entire graph is white.



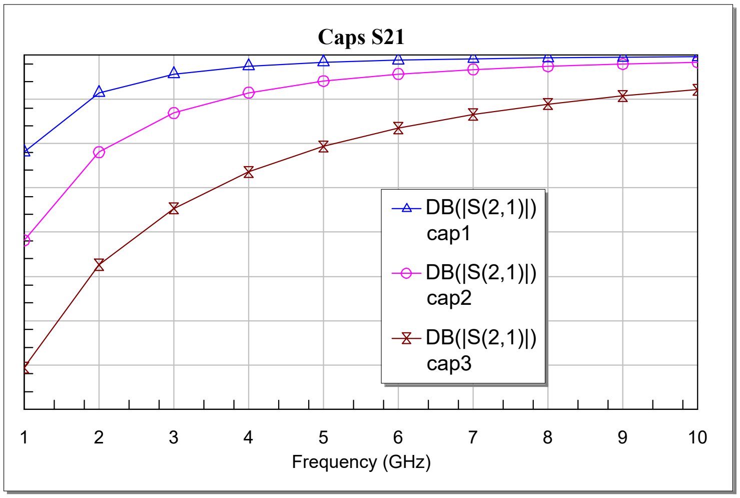
7.1.4.6. Modifying the Graph Division Display

You can manually control many properties of a graph's border and divisions.

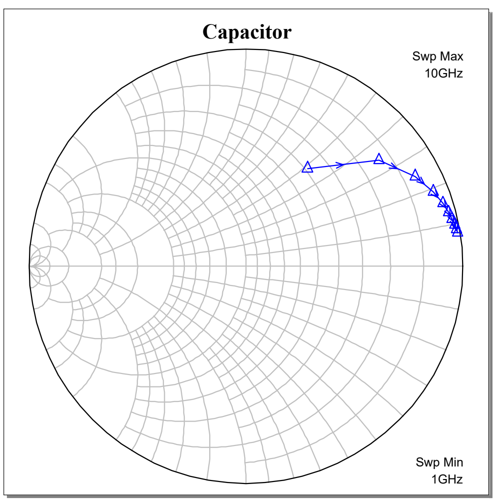
For all graph types, you can change the colors used to display the graph border and divisions. See "Graph Options Dialog Box: Format Tab" for details. The following graph is changed from the previous graph to use blue for the outline and division lines, as well as a thicker line.



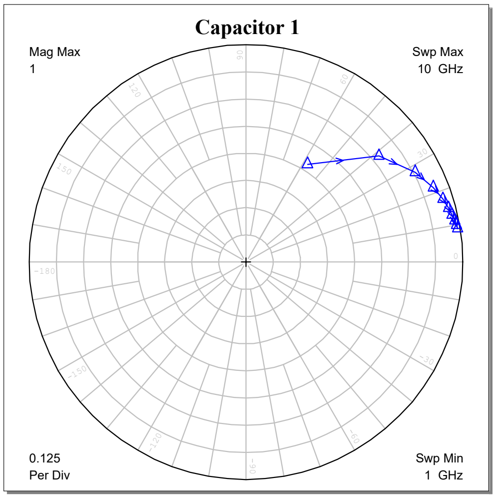
For rectangular graphs, see "Graph Options Dialog Box: Axes Tab" for details of the settings available. You can specify if the values display on the graph, set the number of divisions, and display subdivisions per axis. The following example changes the divisions for the x- and y-axis, turns on subdivisions, and turns off the axis display for the y-axis.



For Smith Charts, see “Smith Chart Options Dialog Box: Grid Tab” for details of the settings available. You can draw impedance and/or admittance contours and specify if the normalized impedance values display on the graph. The following example shows a Smith Chart with admittance contours only, without the impedance values.



For polar grids and antenna plots, see “Antenna/Polar Plot Options Dialog Box: Grid Tab” for details of the settings available. You can specify if the values display on the graph, set the number of divisions for magnitude and angle, and display subdivisions per axis. The following example changes the divisions for the x- and y-axes, turns on subdivisions, and turns off the axis display for the y-axis. The example shows a polar grid with divisions other than the defaults.



7.1.4.7. Data Zooming

There are several options for zooming:

- Zooming on a graph
- Zooming on the graph data
- Changing the axis limits on a graph

Zooming on Graphs

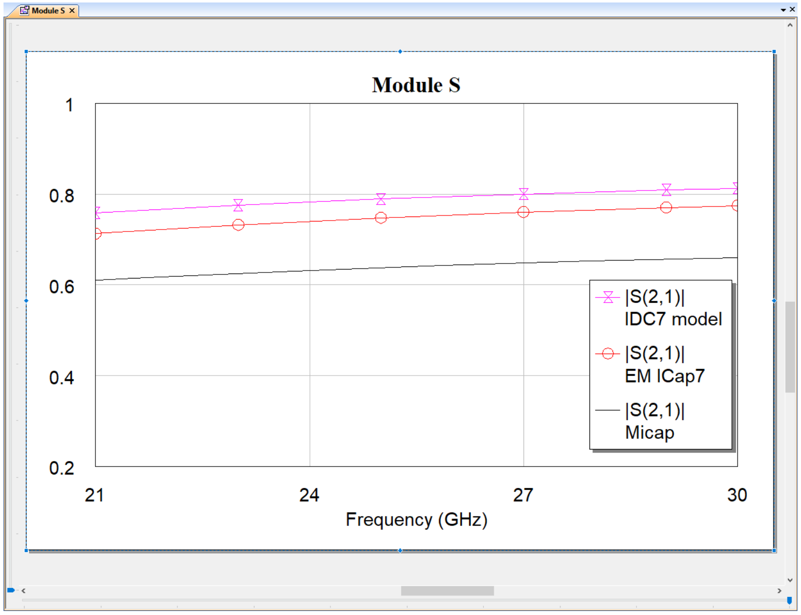
Right-click in any graph to view a menu with zoom commands including:

- **View Area**
- **Zoom In**
- **Zoom Out**
- **View All**

You can also use your mouse to zoom on the graph:

- **Mouse Wheel**: Pans up and down
- **Shift + Mouse Wheel**: Pans left and right
- **Ctrl + Mouse Wheel**: Zooms in and out

Zooming is a quick means of magnifying and panning around a graph, however a zoom is not permanent (when you close and reopen a graph, the zoom level is not saved). The axis displays are not included when zooming. See the following graph zoomed in near the top center of the graph.



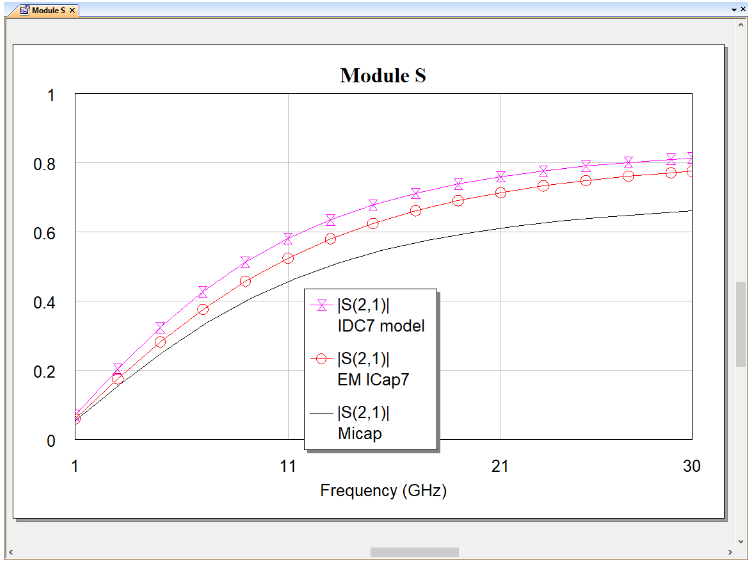
Zooming on Graph Data

The difference between zooming on a graph and zooming on the graph data is that with the data, the axis and legend still display. Like zooming on a graph, the zoom settings are not saved when the graph is closed. If you want to make these settings permanent, you can change the axis settings for the graph. To zoom in on specific data, right-click in a rectangular graph and choose **Zoom Data**, then click and drag the mouse to box in the area you want to magnify. The following table provides guidelines for determining which axis/axes are magnified based upon the area you include in the box you draw.

To Zoom X Axis	To Zoom Left Y Axis	To Zoom Right Y Axis	Box this area:
X	X	X	Inside graph with no axis intersection, or intersecting left y-, right y-, bottom x- (and optionally top x-) axes
X			Intersecting bottom x-axis only, Intersecting top x-axis only, Intersecting top and bottom x-axes only
	X		Intersecting left y-axis only, Intersecting left y- and top x-axes
		X	Intersecting right y- axis only, Intersecting right y- and top x -axes
	X	X	Intersecting both left y- and right y- axes, Intersecting both left y- and right y- axes, and top x-axis

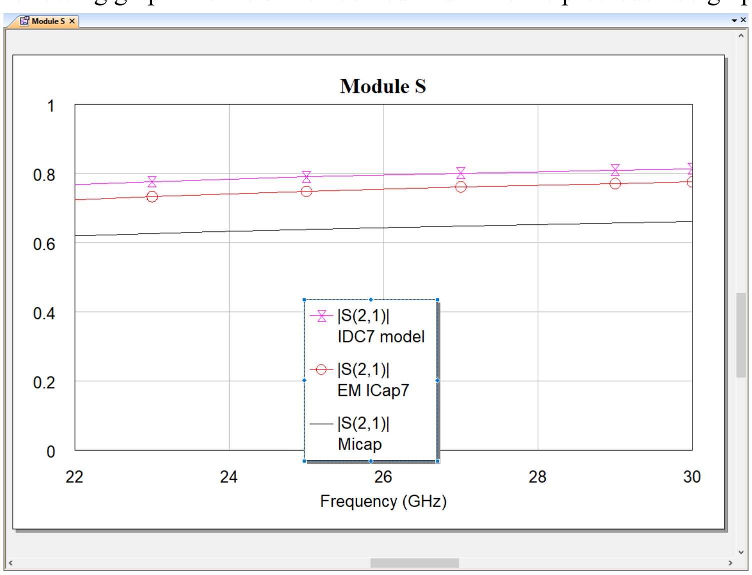
You can easily restore the full range of the graph. To restore the zoom, right-click in a rectangular graph and choose **Restore Axis Settings**.

When you restrict the axis of a rectangular graph, the graph displays sliders to allow you to change the zoomed area. The following graph is zoomed on the data to about the same range as the previous example that zoomed on the graph.

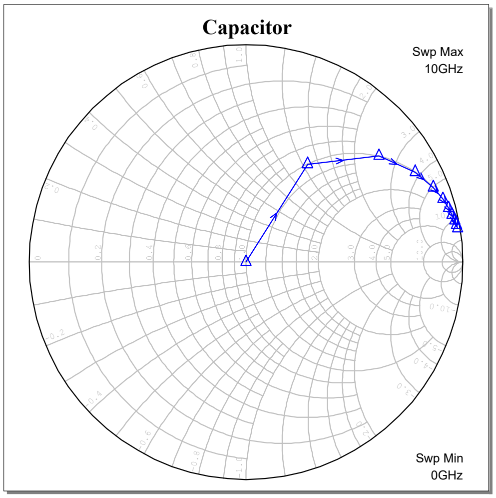


Changing Axis Limits

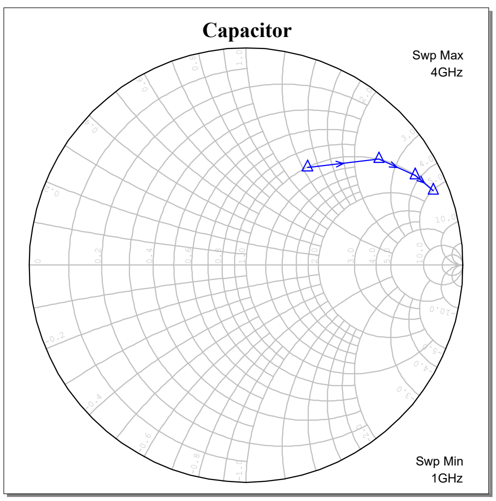
On rectangular graphs, you can permanently change all of the axis limits. The **Limits** section of [“Graph Options Dialog Box: Axes Tab”] controls these settings. You select the axis from **Choose axis** and then apply the desired settings. The following graph has the same zoomed in data as the previous two graphs using the axis settings.



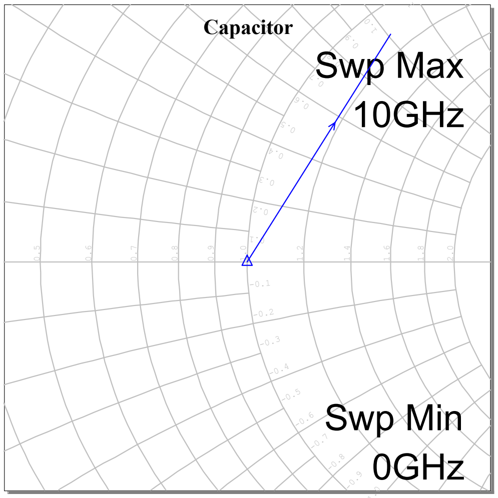
For Smith Charts, there are several control options. See “Smith Chart Options Dialog Box: Grid Tab” for details. The following graph is a Smith Chart with default settings.



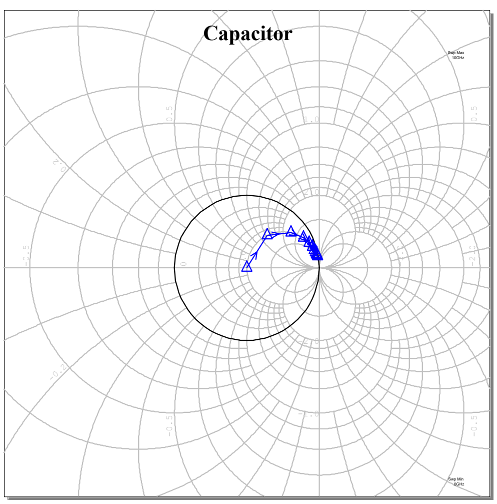
The **Sweep Value Limits** section limits the frequency range in which to display the graph data. The following graph shows the data limited from 1 to 4 GHz.



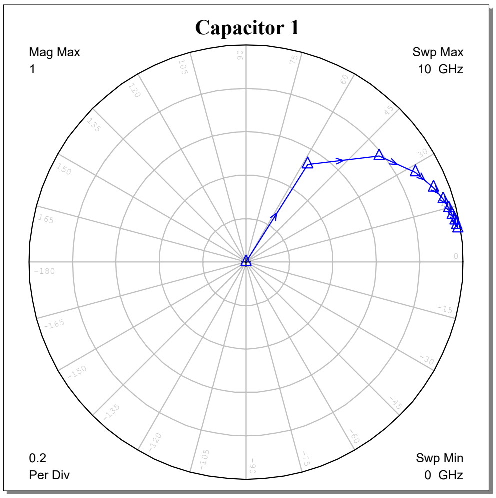
The **Size** section controls how much of a Smith Chart to display. The following graph shows the Smith Chart with an **Expanded** size.



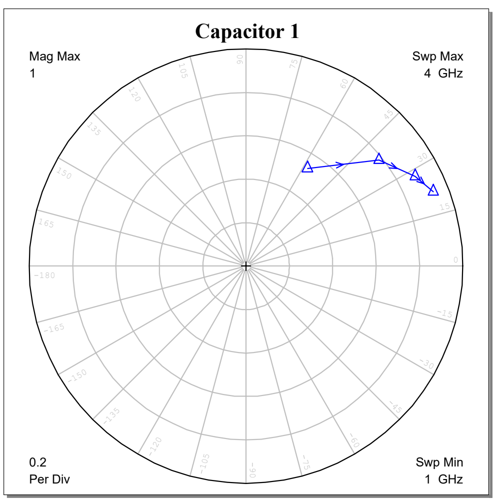
The following graph shows the Smith Chart with a **Compressed** size.



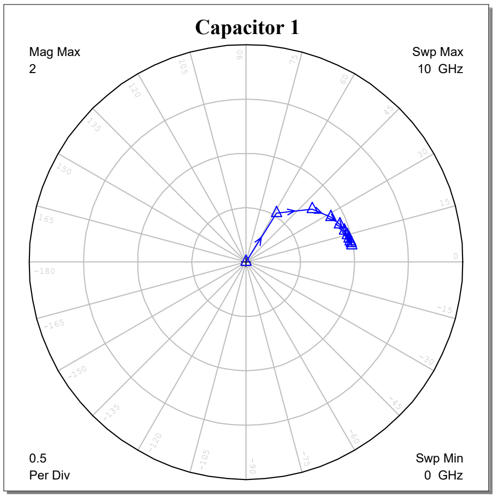
For polar grids and antenna plots, you have more limited control options; see [“Antenna/Polar Plot Options Dialog Box: Grid Tab”]() for details. The following graph is a polar grid with default settings.



The **Sweep Value Limits** section limits the frequency range in which to display the graph data. The following graph shows the data limited from 1 to 4 GHz.



The **Magnitude Limit** section changes the maximum magnitude to display. The following graph shows the polar grid with the maximum limit set to 2.



7.1.4.8. Adding Live Graphs, Schematics, System Diagrams, or Layouts to a Graph

Graphs can also contain other live graphs, schematics, system diagrams, layouts, or 3D views.

To include one of these objects in a graph, simply drag a graph, schematic, or system diagram from the Project Browser to an open graph window. When you release the mouse button a cross cursor displays. Click and drag the cursor diagonally to create a display frame for the added object. When adding a schematic, you can right-click and drag to display a menu with options for inserting it as a schematic, a layout, or a subcircuit. For more information about Window-in-window capabilities see “Window-in-Window ”.

You can also add a shape to a graph by choosing the desired shape type from the **Draw** menu, clicking in the graph window, and drawing the shape.

7.1.5. Copying and Pasting Graphs

The AWR Design Environment platform allows you to copy a graph (including the tabular graph) to the Windows Clipboard, and paste it into another instance of the AWR Design Environment software or into a Windows application such as a word processor, a presentation graphics program, or into the AWR Design Environment platform Design Notes window as part of a project's documentation. See “Creating a New Graph” for information about copying an existing graph to create a new graph.

NOTE: This is a simple way to move a graph from one project to another.

You can also open another view of a graph by choosing **Window > New Window** or by clicking the **New Window** button on the toolbar.

To copy and paste a graph:

1. To copy the entire graph to the Clipboard, choose **Edit > All to Clipboard**, or
2. To copy a zoomed-in area of the graph to the Clipboard, choose **Edit > View to Clipboard**. The area of the picture copied to the Clipboard is determined by the border of the window in which the graph displays.
3. Paste the results into the destination program.

7.2. Working with Measurements

A measurement is data such as gain, noise, power, or voltage that is computed by a simulation and plotted on a graph (or otherwise output). Every measurement is associated with a particular graph, and it displays as a subnode of that graph in the Project Browser. When you choose **Simulate > Analyze** to perform simulations, the required simulator is invoked for each particular measurement.

The measurements that simulations can compute are organized into categories (measurement types). For an overview of the categories and detailed descriptions of Microwave Office measurements, see the AWR Microwave Office Measurement Catalog. For an overview of the categories and detailed descriptions of VSS measurements, see the AWR Visual System Simulator Measurement Catalog.

Note that measurements transform N-port data sources into a vector of real or complex data that can be plotted on a graph.

You can list and modify measurements associated with a graph by right-clicking in the graph window or the graph legend and choosing **Modify Measurement** to display the Modify Measurement dialog box.

7.2.1. Adding a New Measurement

You can add a new measurement from the Project Browser or from another source such as a schematic, system diagram, EM document, or output equation.

7.2.1.1. Adding a Measurement from the Project Browser

To add a new measurement from the Project Browser:

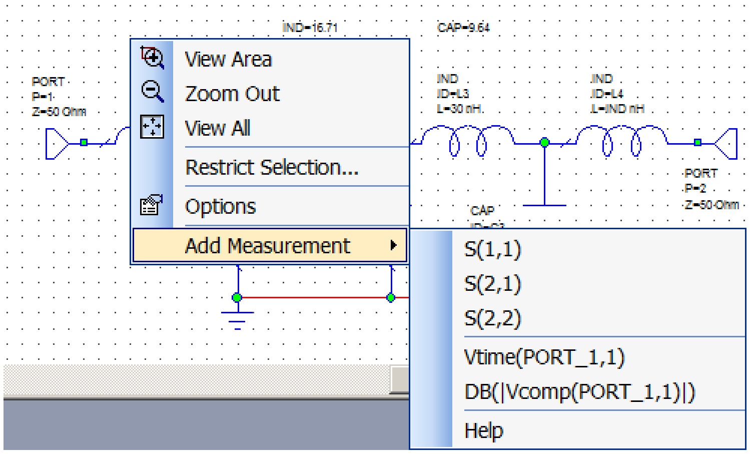
1. Right-click a graph node in the Project Browser and choose **Add Measurement**, or choose **Project > Add Measurement**.

   The Add Measurement dialog box displays. See “Add/Modify Measurement Dialog Box” for more information about this dialog box.
2. Select the desired **Measurement Type** and **Measurement**, specify the desired options, and click **OK**. For comprehensive information about all available measurements, see the AWR Microwave Office Measurement Catalog or the AWR Visual System Simulator Measurement Catalog.

   The Project Browser displays the new measurement under the target graph. The name of the new measurement conforms to the standard measurement naming conventions.

7.2.1.2. Adding a Measurement through Another Source

To add a measurement to a schematic, system diagram, or EM document, right-click in the document and choose **Add Measurement**. A submenu displays with a list of previously selected favorite measurements, as shown in the following figure.



You can specify a measurement as a favorite by clicking the **Favorite** button while the measurement is selected in the Add/Modify Measurement dialog box. Favorites display in the submenu only in a new AWR Design Environment program session. After choosing a favorite measurement from the submenu, a Select graph for new measurement dialog box displays to prompt you to select the graph to which you want to add the measurement. You can also click **New Graph** to add the measurement to a new graph. After selecting the graph, the Add/Modify Measurement dialog box displays to allow you to edit the measurement if needed. For example, you can change the port index or test point. To suppress the graph prompt, **Ctrl**-right-click to select a measurement from the submenu. To suppress the display of the Add/Modify Measurement dialog box, **Shift**-click the **OK** button in the Select graph for new measurement dialog box. When you suppress the graph prompt, the first graph available is used. If no graphs exist, a new rectangular graph is created and used. After you make your selections, the specified graph opens and the measurement is placed on it. You can now simulate to see the new results.

When plotting on a real-valued graph, the AWR Design Environment platform auto-converts complex measurements to real measurements using the DB-magnitude complex modifier. Conversely, when plotting on a graph that supports complex results, any complex modifier is removed. This ability means that you only need one entry, for example, “S(1,1)” that plots as S(1,1) on a Smith chart and as DB(|S(1,1)|) on a rectangular graph.

You can also add measurements from the source in Output Equations. You do not need to specify a list of favorite measurements since the measurement is simply `Eqn(var_name)`.

7.2.1.3. Measurement Naming Conventions

The names of measurements displayed in the Project Browser are composed of two parts. The first part is the name of the data source the measurement uses. The second part is the measurement type being created. The two parts of the name are separated by a colon (:). An example measurement name is:

MySchematic:|Icomp(DCVS.Vcollector,0)|[*,X]

where MySchematic is the data name, and Icomp is the measurement type. Depending on the type of measurement chosen, various properties of the measurement may display as arguments in parentheses. This specific case indicates that the current Icomp is measured in a DCVS element whose ID is Vcollector. The harmonic number of this specific measurement is specified in the second argument and has the value 0. The vertical bar symbols denote that the magnitude of the current Icomp is specified. When swept parameters are used, the sweep arguments display in square brackets at the end of the measurement name. In this case, the measurement has two swept parameters. For the first swept parameter, the "*" indicates that all values of the swept parameter display. For the second swept parameter, the "X" indicates that this parameter displays on the x-axis of the graph. The data source name "All Sources" is reserved for template measurements, as discussed in “Using Project Templates with Template Measurements”.

7.2.1.4. Ordering Measurements

To order a new (copied) measurement amongst existing measurements, drop it on top of the measurement above which you want it to display. To place it at the bottom of the list of measurements, drop it onto the graph node.

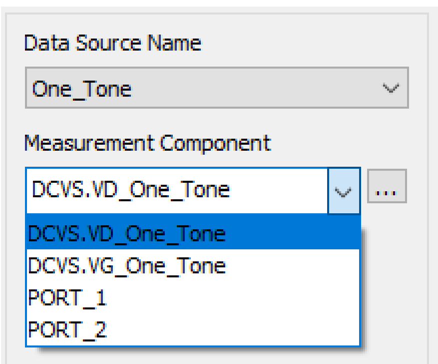
To reorder existing measurements, select the measurement you want to move and press **Alt + Up Arrow** or **Alt + Down Arrow** to move the measurement accordingly. Alternatively, simply drag the measurement to the position you want. The graph legend also reflects this revised order of measurements.

7.2.2. Measurement Location Selection

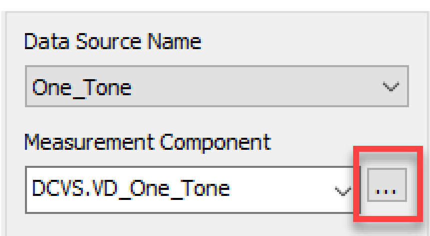
You can make linear measurements only at ports, while you can make nonlinear and system (VSS) measurements at any node in the circuit. When adding nonlinear and system measurements for circuit analysis, the **Measurement Component** you select in the Add/Modify Measurement dialog box includes any ports and sources by default.

NOTE: For nonlinear circuit simulation, you can use the M_PROBE in a schematic and point measurements to the probe as an easy way to probe nodes in a circuit. See "Implementation Details" for more information.

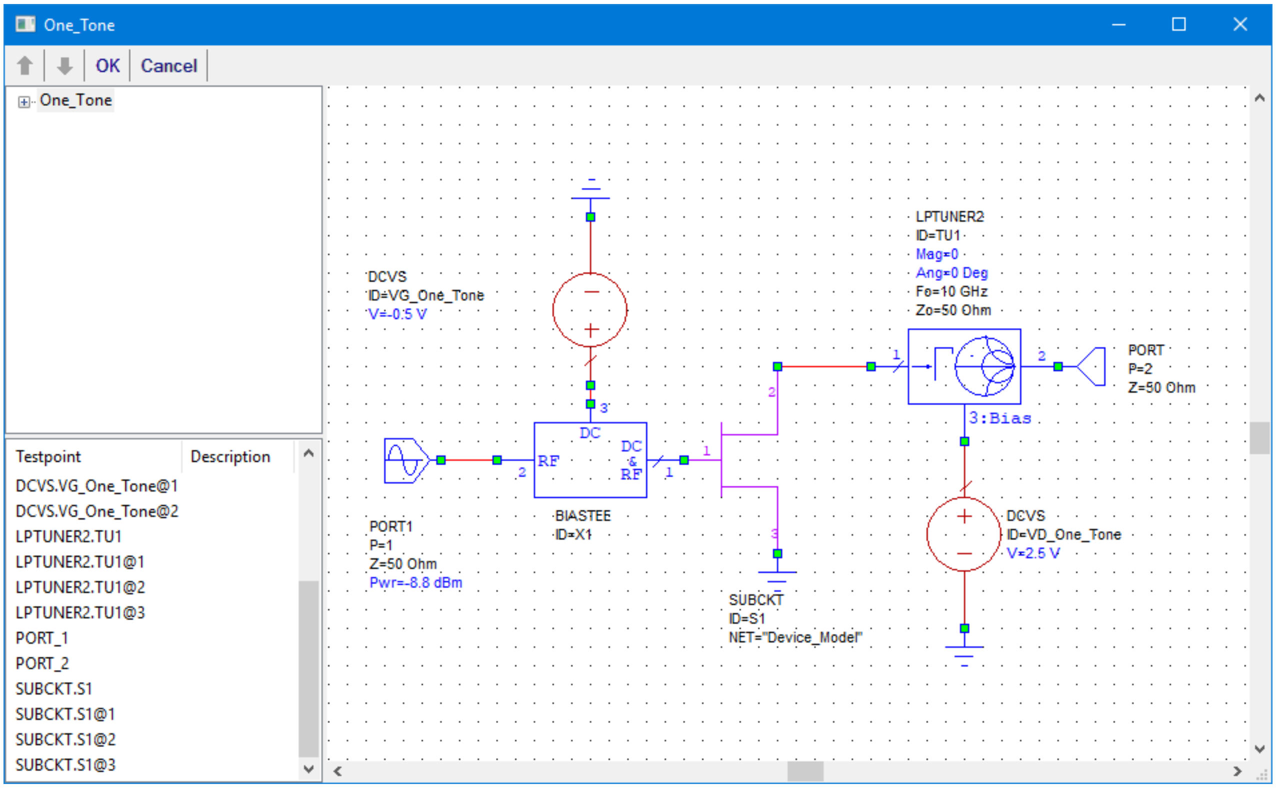
For example, the following nonlinear measurement for a circuit has two ports and two DC sources.



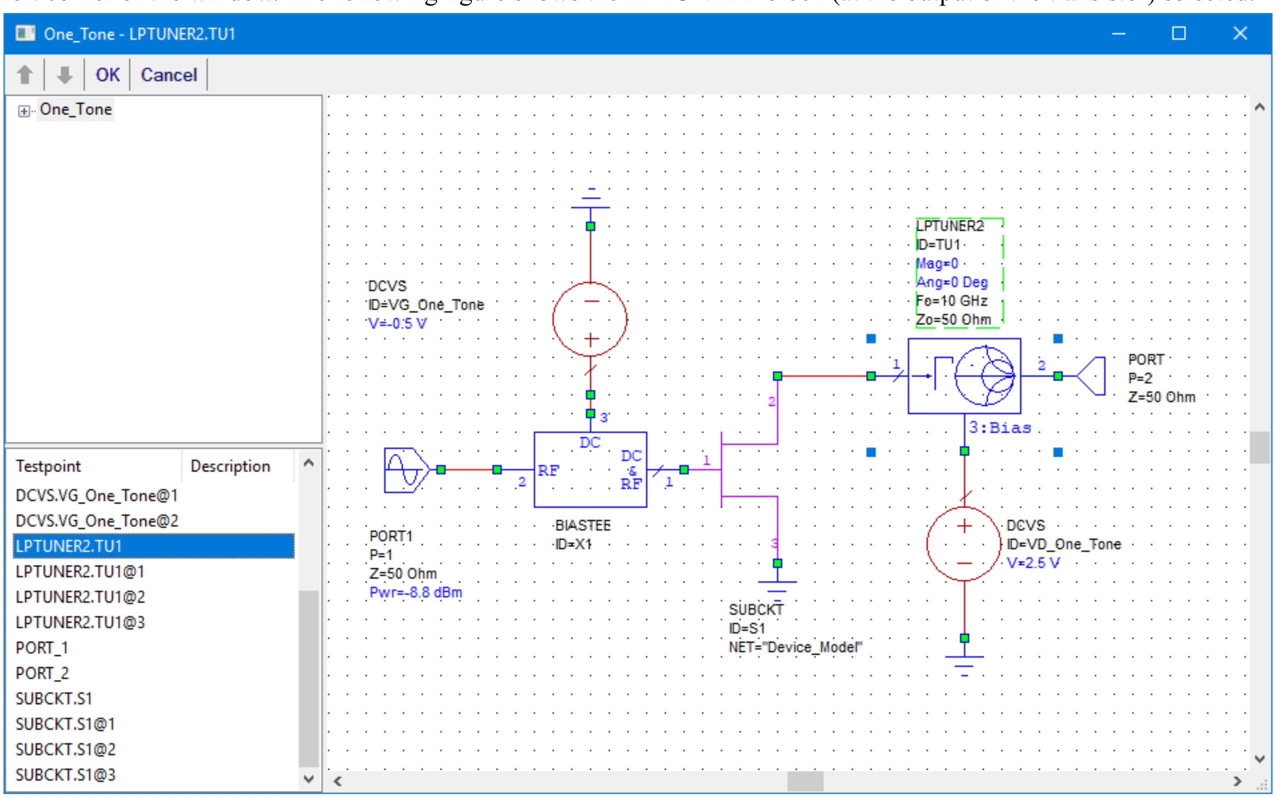
You can click the ellipsis button to display a window that allows you to choose any node in your circuit to perform your measurement.



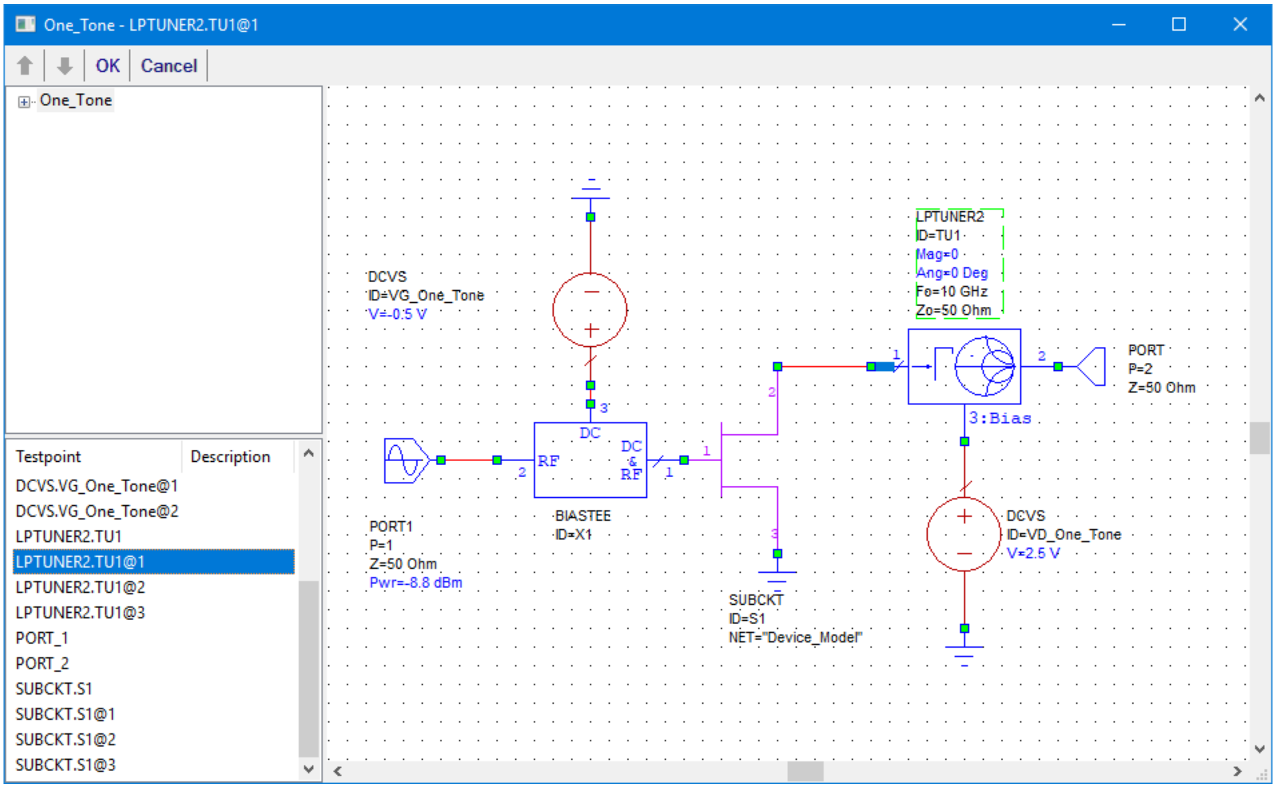
The window is a view of the schematic specified in **Data Source Name**.



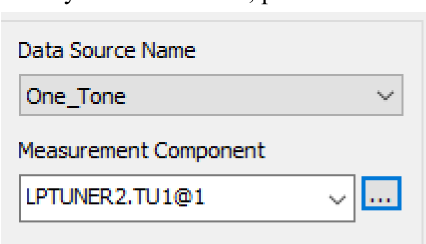
You can select the component where you want the measurement made; this model name displays in **Testpoint** in the lower left corner of the window. The following figure shows the LPTUNER2 block (at the output of the transistor) selected.



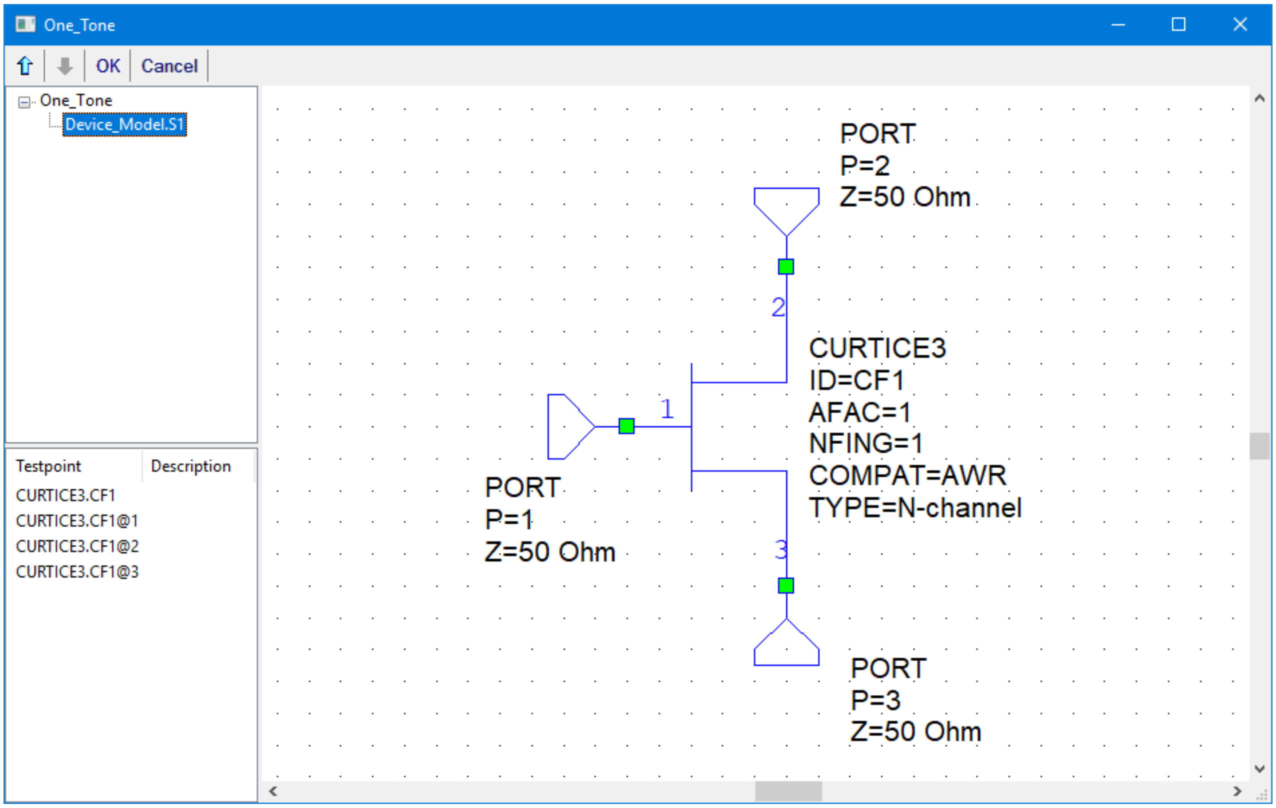
The model itself is not enough information; in **Testpoint**, you need to select the proper node number for the model. These items are listed below the selected model using @N syntax, where N is the node number. The following figure shows node 1 selected for this model.



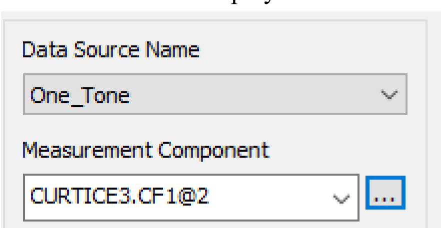
After you select a node, press **Enter** or click **OK** at the top left of the window.



You can also use this window to select locations down through the schematic hierarchy. At the upper left you can expand the name of the top level schematic to show any subcircuit instances. Click on an instance name to display it in the window. The following figure shows the subcircuit selected in this example.



The selected node displays as the **Measurement Component** in the Add/Modify Measurement dialog box.



You can transverse hierarchy directly in the view of the schematic by selecting any subcircuit, right-clicking and choosing **Edit Subcircuit**, or clicking the down arrow button at the top of the window. You can push back up through hierarchy by right-clicking with nothing selected and choosing **Exit Subcircuit**, or clicking the down arrow button at the top of the window.

NOTES: Some nonlinear models can also have measurements made across internal branches of the model. These branches display in this window.

When the measurement component of a current measurement is an element without a node number, the measurement result is the current into node 1 of the element.

7.2.3. Modifying, Copying, and Deleting Measurements

You can modify, copy, or delete the measurements associated with any graph, as well as specify that obsolete measurements continue to display (in gray). To view and edit the current project's measurements individually or in collections, you can also use the Measurement Editor.

7.2.3.1. Modifying Measurements

To modify a measurement:

1. Double-click the desired measurement in the Project Browser, or right-click the measurement in the graph legend and choose **Modify Measurement**. The Modify Measurement dialog box, which is identical to the Add Measurement dialog box, displays. See "Add/Modify Measurement Dialog Box" for more information about this dialog box.
2. Make the desired modifications, and click **OK**.

7.2.3.2. Copying Measurements

To copy a measurement from one graph to another, select the measurement in the Project Browser and drag and drop it on the target graph node. When compatible, the AWR Design Environment platform automatically converts the measurement for the new graph type. See "Ordering Measurements" for information about ordering the measurement amongst others in the graph.

To copy a measurement on the same graph, select the measurement in the Project Browser and drag and drop it on the same graph node. The Add/Modify Measurement dialog box automatically displays.

7.2.3.3. Deleting Measurements

To delete a measurement, do one of the following:

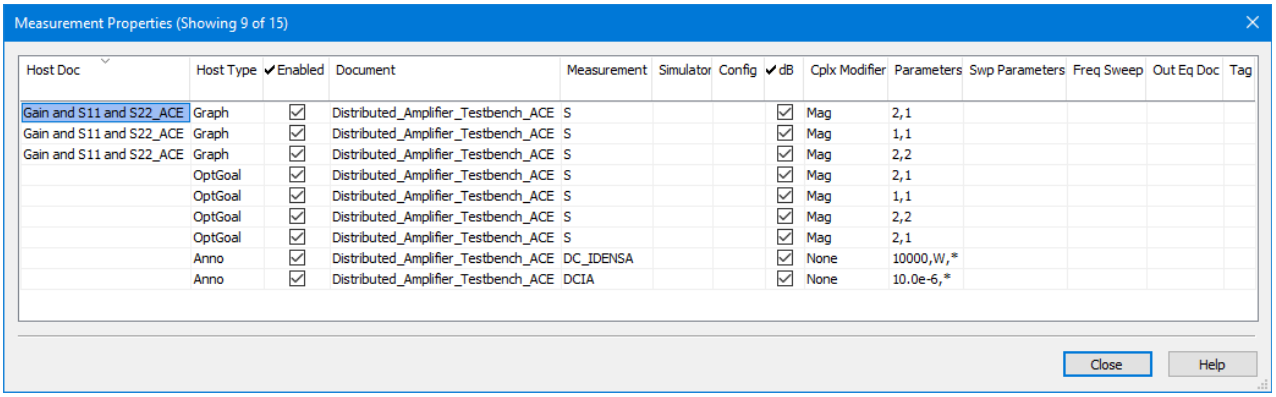
- Select the measurement in the Project Browser, and choose **Edit > Delete**, or
- Right-click the measurement in the Project Browser, and choose **Delete** , or
- Select the measurement in the Project Browser, and press the **Delete** key.

7.2.3.4. Displaying Obsolete Graph Measurements

Measurement results are plotted on a graph (in a dimmed color) even when the simulated graph data is outdated due to user changes in component values or geometries, or because of changes in equations that affect the measurement results. The associated graph legend entry also displays in gray. A graph can contain a mixture of active and obsolete plot and legend entries as appropriate for the individual measurement status.

7.2.4. Using the Measurement Editor

The Measurement Editor displays the active project measurements with their options, allowing you to quickly edit measurements individually or in collections. The Measurement Editor provides various ways to manage the measurements in your project, such as changing their sort order, filtering on specific criteria, editing multiples, and pre-populating the filter selection by choosing to open the Measurement Editor off of a particular source.



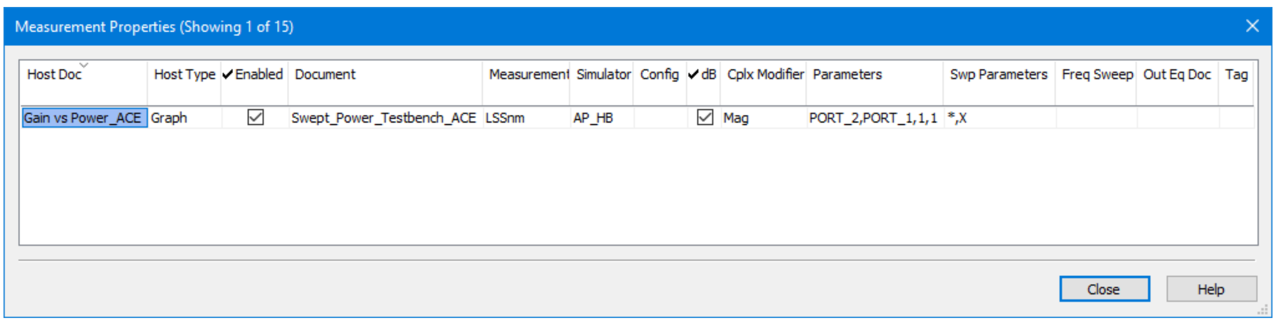
Measurement Properties (Showing 9 of 15)

Host Doc	Host Type	✔Enabled	Document	Measurement	Simulator	Config	✔dB	Cplx Modifier	Parameters	Swp Parameters	Freq Sweep	Out Eq Doc	Tag
Gain and S11 and S22_ACE	Graph	☑	Distributed_Amplifier_Testbench_ACE	S			☑	Mag	2,1				
Gain and S11 and S22_ACE	Graph	☑	Distributed_Amplifier_Testbench_ACE	S			☑	Mag	1,1				
Gain and S11 and S22_ACE	Graph	☑	Distributed_Amplifier_Testbench_ACE	S			☑	Mag	2,2				
	OptGoal	☑	Distributed_Amplifier_Testbench_ACE	S			☑	Mag	2,1				
	OptGoal	☑	Distributed_Amplifier_Testbench_ACE	S			☑	Mag	1,1				
	OptGoal	☑	Distributed_Amplifier_Testbench_ACE	S			☑	Mag	2,2				
	OptGoal	☑	Distributed_Amplifier_Testbench_ACE	S			☑	Mag	2,1				
	Anno	☑	Distributed_Amplifier_Testbench_ACE	DC_IDENSA			☑	None	10000,W,*				
	Anno	☑	Distributed_Amplifier_Testbench_ACE	DCIA			☑	None	10.0e-6,*				

Close Help

To open the Measurement Editor, right-click a source document such as a schematic, EM structure, or system diagram, or a host document such as a graph, optimization goal, or annotation, and choose **Edit Measurements**. You can also right-click a graph in the Project Browser and choose **Edit Measurements** or choose **Graph > Edit Measurements**. See [“Measurement Editor Columns”](#) for a list of supported source and host documents.

When you open the Measurement Editor in this manner, it automatically applies a filter for the source or host document. For example, if you open the Measurement Editor from a schematic named "Swept_Power", a filter is applied in the **Document** column for the "Swept_Power" schematic.



Measurement Properties (Showing 1 of 15)

Host Doc	Host Type	✔Enabled	Document	Measurement	Simulator	Config	✔dB	Cplx Modifier	Parameters	Swp Parameters	Freq Sweep	Out Eq Doc	Tag
Gain vs Power_ACE	Graph	☑	Swept_Power_Testbench_ACE	LSSnm	AP_HB		☑	Mag	PORT_2,PORT_1,1,1	*,X			

Close Help

NOTE: The Measurement Editor does not restrict access to specific settings for each measurement the way the Add/Modify Measurement dialog box does. For example, the Vtime (time domain voltage) measurement has real values, but the Measurement Editor allows checking the box under **dB** or setting **Cplx Modifier**. Use caution when editing multiple measurements to ensure the modified settings are applicable to the measurements.

7.2.4.1. Navigating the Measurement Editor

The Measurement Editor allows you to edit field entries singly or in multiples, by **Ctrl**-clicking each item. You can also **Shift**-click to select consecutive items in a single column, or press **Ctrl + A** to select the entire column. Multi-selection works within a single column only; it does not span columns. You can maintain the row selection when navigating to a different column, however. After multi-selecting you can press the **Shift** key and then use the left and right arrow keys to select the same rows in adjacent columns.

To customize the width of a Measurement Editor column you can drag the column's right boundary. To move a column, you can click on the column header and drag it to a different position.

7.2.4.2. Measurement Editor Columns

Host Doc - displays the measurement location (for example, the graph name). This field is read only.

Host Type - displays the **Host Doc** type. Valid types are Graph, OptGoal (Optimization Goal), YldGoal (Yield Goal), Anno (Annotation), (OutFile) Output File, or OutEqn (Output Equation). This field is read only.

Enabled - selected indicates that the measurement is enabled.

Document - the source document that the measurement is using for its data. You can select from a list of sources: circuit schematics, EM structures, system diagrams, data files, or output equation documents.

Measurement - type the name of the measurement you desire to make on the source document. If the string you enter is not a valid measurement it retains its original value. This field auto-corrects case.

Simulator - the simulator used in the measurement. There are two-letter shortcuts for each non-default simulator; which is leaving the field blank. In addition to the following shortcuts, you can enter any of the other simulators in the **Simulator** column that apply for the measurement being made:

- Blank - default simulators
- AP - APLAC linear or APLAC HB, depending on the source document.
- AP_TR - APLAC transient
- SP - Spectre

Config - the Switch List to use.

dB - selected indicates that the measurement is plotted in dB.

Complex Modifier - the modifier applied to the simulation data. You can select from the following modifiers: None, Real, Imag, Mag, Ang, Angu, or Conj.

Parameters - a comma separated list of measurement parameter values that are meaningful to a measurement, that are listed in the parentheses of the measurement string. For an S-parameter measurement, for example, the parameters are the ports, and are entered as **1,1** for **S(1,1)**. You can enter this list with or without parentheses. You should have knowledge of what comprises a measurement string before editing the parameters of a measurement.

Sweep Parameters - a comma separated list of the measurement sweep parameter values that reside in the brackets of the measurement string. Controls how the sweeps (frequency or otherwise) display on the graph. You should have knowledge of what comprises a measurement string before editing the sweep parameters of a measurement. If the Sweep Parameters are completely blank then the first sweep (usually frequency) is used for the X axis and all traces are plotted for the other sweeps. The valid sweep parameter entries are:

- X - use for X axis
- "*" - plot all traces
- Integer - specifies a particular one-based index
- ~ - disable Sweep
- T - select with Tuner

Frequency Sweep - the frequency list that the measurement is plotting. The supported frequency lists, where applicable, are: FDOC, FPRJ, F_OSC, FSAMP, FSPEC, FDOCN, F_DC, and F_SYMB. You may also specify a frequency list from a sweep frequency block by typing the ID of the sweep frequency block (for example, type **FSWP1** if the sweep frequency block is SWPFRQ.FSWP1). When this entry is left blank the frequencies used are FDOC.

Tag - The tag for the measurement. You can use tags for grouping and filtering of measurements in the Measurement Editor.

7.2.4.3. Sorting and Filtering

You can sort the measurements in the Measurement Editor in any column in ascending or descending order by clicking that column header, and clicking again to reverse the sort order. You can also sort on multiple columns by clicking the column header of the first column to set the sort order for that column and then clicking the second column header to set the sort order of that column.

In addition to sorting you can also filter to find a specific measurement or set of measurements. Filtering is enabled for every column. To filter on a column, click in the filter text box below the column name and type the text you want to filter for in that column. For example, to find all S-parameter measurements in your project you can type "**S**" in the filter text box in the Measurement column.

The filter text box also supports regular expressions, increasing the ability to perform intelligent searches. The form and functionality of these regular expressions is modeled after the regular expression facility in the Perl 5 programming language. The following table shows some syntax examples.

Syntax	Comment
.	Match any single character
*	Match zero or more of the preceding characters
+	Match one or more of the preceding characters
?	Match zero or one of the preceding characters
!	Filter out subsequent characters
\d	Match any digit (0-9)
ch[at]	Match cat and hat
W[1-3]	Match W1, W2, and W3
^M	Match names that start with M
^W\d+	Match names that start with W followed by one or more digits
\$$	Match names that end in $

7.2.4.4. Tagging

You can enter one or more user-defined tags in the **Tag** column to associate measurements with that phrase. For example, if you are plotting power curves you might enter the bias of the transistor as the tag to remind yourself which measurement is for what bias.

Filtering is accomplished with a sub-string search that displays all measurements that contain a tag that matches the filter text. If tagging is set up properly, filtering and sorting with tags provide a great way to keep the measurements in your design organized. For information on filtering and sorting, see “Sorting and Filtering”.

7.2.5. Disabling a Measurement from Simulation

To prevent a measurement from being computed when you choose **Simulate > Analyze**, you can disable the measurement.

To disable/enable individual measurements, right-click on the measurement and choose **Toggle Enable**. When one or more measurements are disabled, you can right-click on the associated graph node and choose **Toggle All Measurements** to reverse the disabled/enabled status of *all* measurements.

To disable all measurements in a graph, right-click the associated graph node in the Project Browser and choose **Disable All Measurements**. You can re-enable all measurements by choosing **Enable All Measurements**.

To disable all measurements in a project, right-click the **Graphs** node in the Project Browser and choose **Disable All Measurements**. You can re-enable all measurements by choosing **Enable All Measurements**.

7.2.6. Simulating Only Open Graphs

To simulate only the open graphs in your project, right-click the **Graphs** node in the Project Browser and choose **Simulate Open Graphs**.

7.2.7. Post-Processing Measurements and Plotting the Results

You can use the Output Equation feature of the AWR Design Environment platform to assign the result of a measurement to a variable. You can then use this variable in other equations just like any other variable, and you can plot the final "post-processed" result just like any other measurement.

For information on defining variables and equations for this purpose, see "Assigning the Result of a Measurement to a Variable". For information on how to plot the final result, see "Plotting Output Equations".

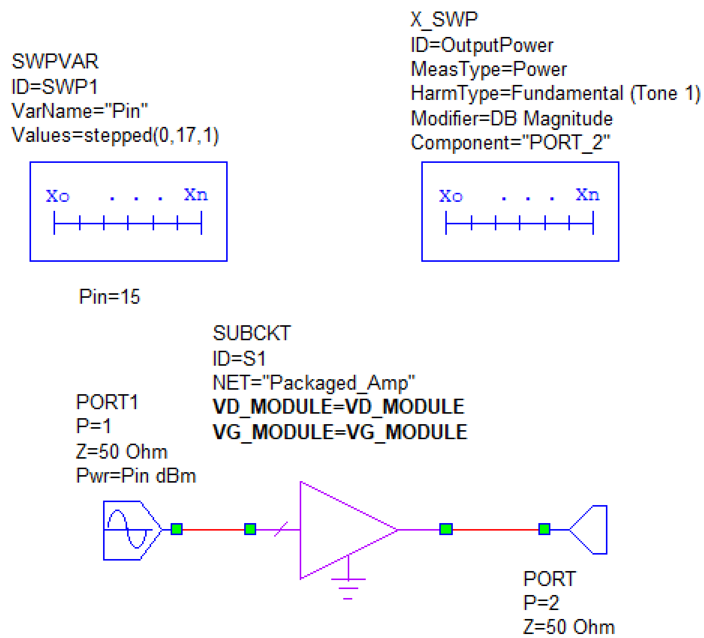
7.2.8. Measurements with Swept Variables

When you define sweeping (frequency, power, etc.), the Add/Modify Measurement dialog box controls how the swept analysis data displays.

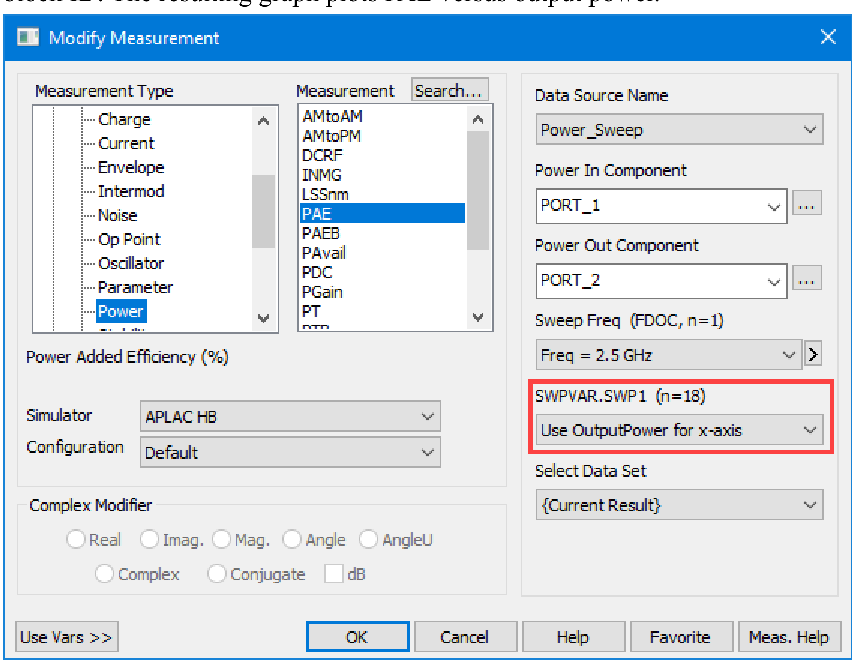
For information on swept variable analysis, see "Swept Parameter Analysis ".

7.2.9. Plotting One Measurement vs. Output Power, Voltage, or Current

You can plot a measurement versus output power, voltage, or current, instead of a swept input quantity. To specify a user-defined x-axis for a measurement, place an X_SWP block in the schematic on which you are making the measurement. On the X_SWP block, specify the x-axis quantity type (power, voltage, or current), and the component node on which the x-axis quantity is measured. In the following figure, the input power is swept using the SWPVAR (ID = SWP1) block. The X_SWP (ID = OutputPower) block is set up to measure the fundamental output power at Port 2.



When you make a measurement on a schematic with an X_SWP block, the X_SWP block ID displays as an x-axis drop-down option for swept parameters in the Add/Modify Measurement dialog box. The following figure shows the dialog box corresponding to a PAE measurement on the schematic shown in the previous figure. For the SWPVAR.SWP1 parameter, **Use OutputPower for x-axis** is selected instead of **Use for x-axis**, where **OutputPower** corresponds to the X_SWP block ID. The resulting graph plots PAE versus output power.



You can add multiple X_SWP blocks to a schematic to plot measurements versus various harmonic power, voltage, and current components measured at various nodes. See the "User-defined X-axis Value Sweep: X_SWP " for more information about this block.

7.2.10. Plotting One Measurement vs. Another Measurement

You can plot one measurement versus another measurement. Typical measurements have an input sweep on the x-axis (for example, frequency or input power). If you want to put a measurement other than power, voltage, or current on the x-axis, you can use the PlotVs measurement in the **Data** measurement category. See "Plot Measurement 1 vs Measurement 2: PlotVs" for more information about this measurement. If you want power, voltage, or current for the x-axis, use the X_SWP element instead of the PlotVs measurement.

7.2.11. Single Source vs. Template Measurements

Template measurements are Microwave Office measurements you create by choosing **All Sources** as the **Data Source Name** in the Add Measurement dialog box. A template measurement creates a measurement for each data source that is added to the project. When a data source is removed from the project, the measurements for the source that were created from measurement templates are also removed. Measurement templates provide a method for specifying a particular measurement that is to be made for each of the data sources in the project, without creating individual measurements for each data source.

A measurement that is associated with a particular data source is a single source measurement. Single source measurements are created by selecting the name of the associated data source as the **Data Source Name** in the Add Measurement dialog box. Since single source measurements reference a particular data source, if the data source is deleted or renamed, the measurement generates an error.

7.2.12. Using Project Templates with Template Measurements

You can use project templates to save options, LPFs, artwork cells, design notes, global definitions, frequency, graph, and measurement information for a particular project for use in other projects or for comparison purposes. When you create a project template, it saves all frequency and graph information, and all measurements that are specified as having **All Sources** as the **Data Source Name**.

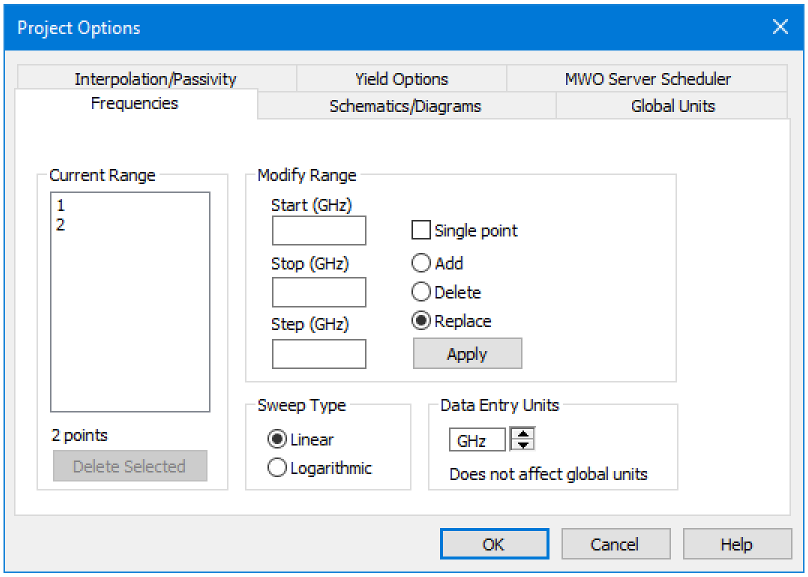
You can use project templates with template measurements to allow the Microwave Office program to be used as a default viewer for N-port data sources. For example, if the Microwave Office program is associated with sources that have a **.s2p* extension, then if you click a *mysource.s2p* source in the Windows source manager or Explorer, the Microwave Office program loads the default project template and adds the *mysource.s2p* source to the project. If the default project template includes template measurements, the measurements of the *mysource.s2p* source are automatically created, and the desired measurements of *mysource.s2p* automatically display.

7.2.12.1. Measurement Comparison Using Project Templates

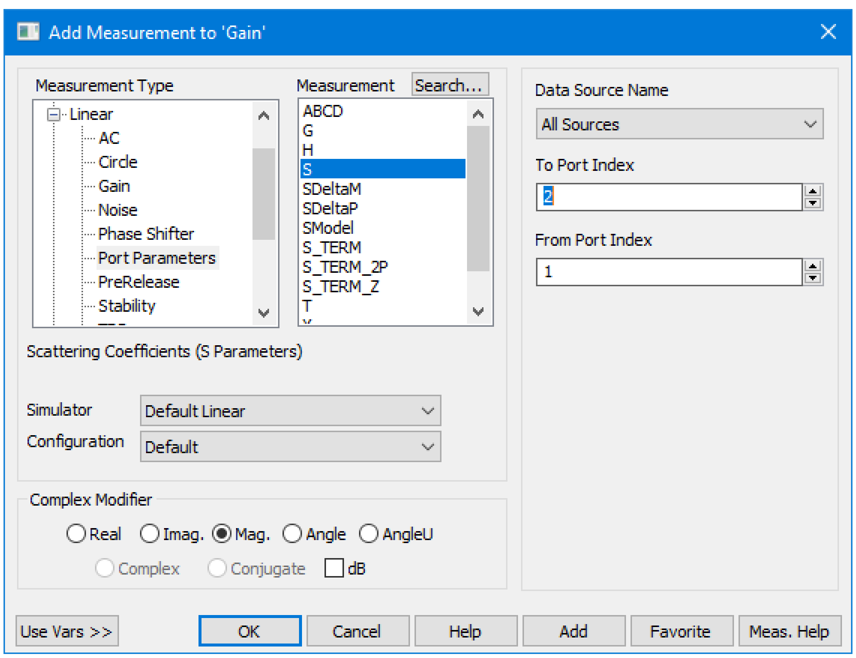
Project templates are useful for comparing measurements of various data files. The following example illustrates this utility.

In this example, S-parameter data files are compared for gain and return loss over a frequency range of 2-18 GHz. The first step is to create a project template that includes the frequencies, graphs, and measurements required for the comparison. To create the template:

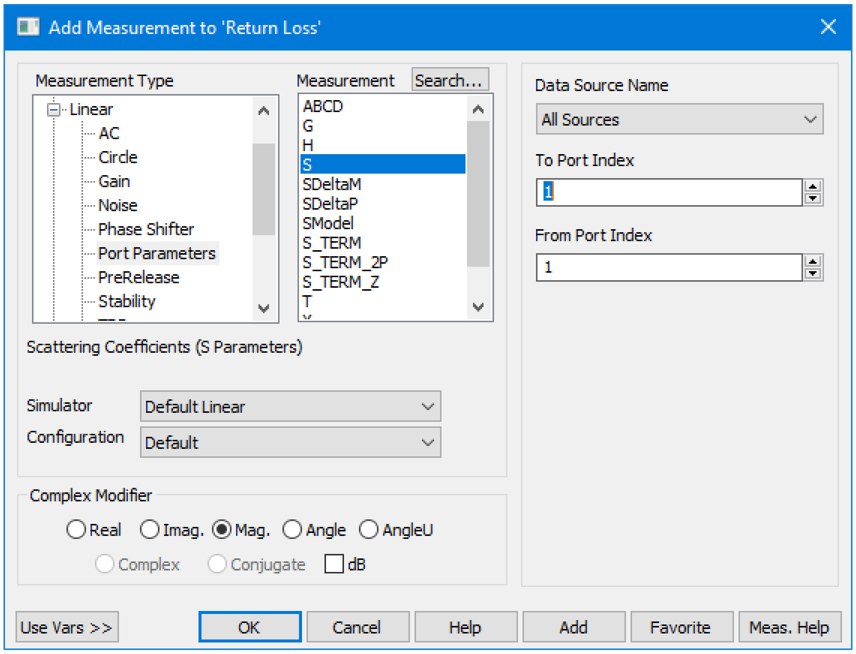
1. Double-click **Project Options** in the Project Browser. In the Project Options dialog box on the **Frequencies** tab, specify the following values and click **Apply**.



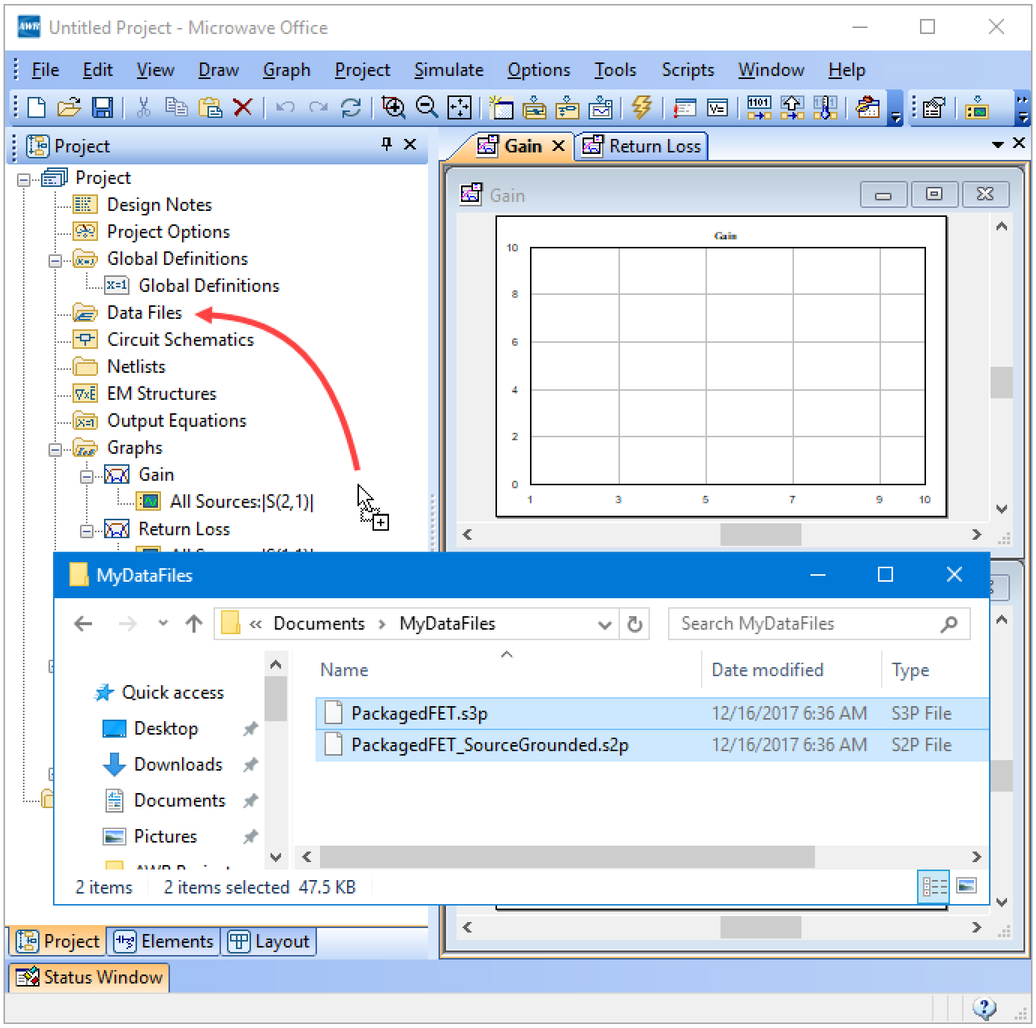
2. Right-click **Graphs** in the Project Browser and add a rectangular graph named "Gain". Repeat the same step to add a rectangular graph named "Return Loss".
3. Right-click the "Gain" graph and choose **Add New Measurement**. Add a measurement with the following values and click **OK**.



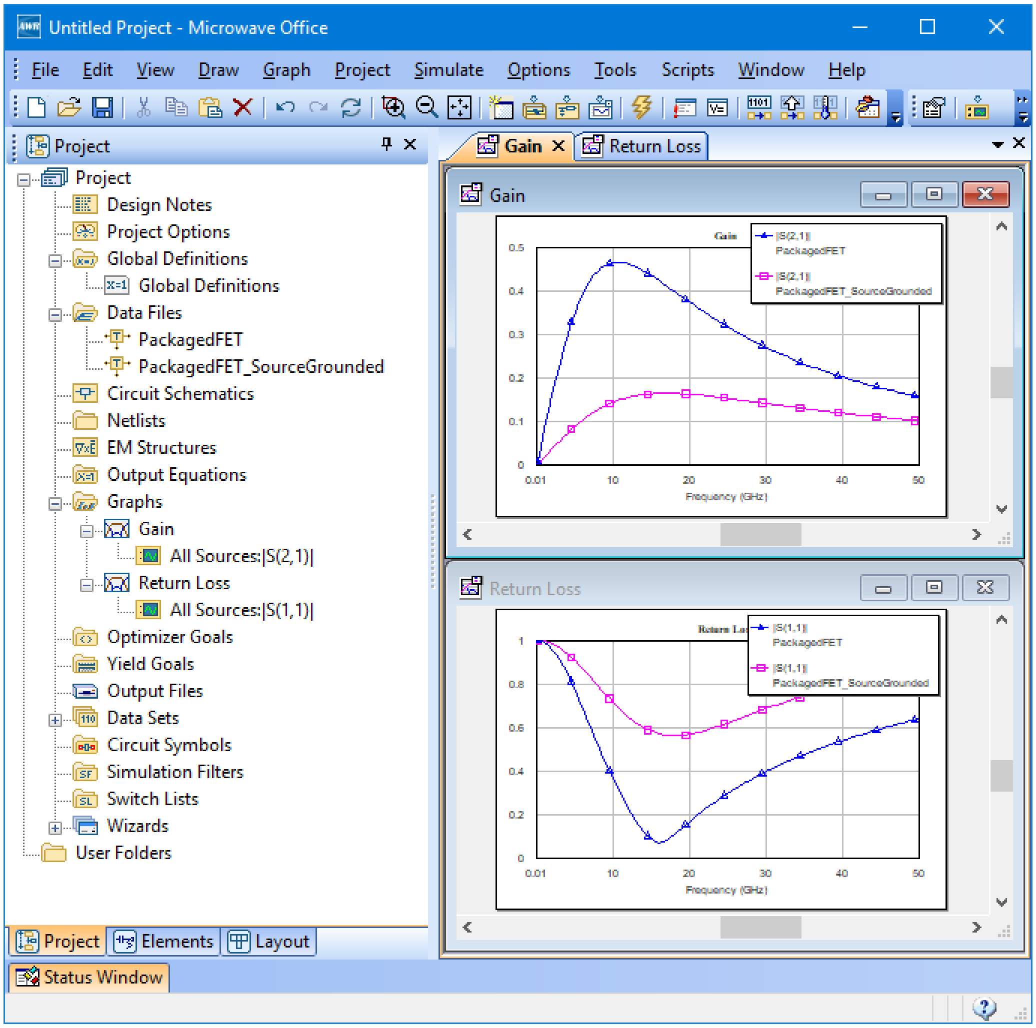
4. Right-click the "Return Loss" graph and choose **Add New Measurement**. Add a measurement with the following values and click **OK**.



5. To save the measurements, graphs, and frequencies in a project template, choose **File > Save Project As**. In **Save As type**, choose **Project Template (*.emt)**. Name the file "Compare data" and click **Save**.
6. To compare S-parameter data files, open Windows Explorer to view the data files. Drag and drop the files onto the Project Browser **Data Files** node.



The measurements in the template automatically display on the graphs after simulation, as shown in the following figure.



7.3. Working with Output Files

In addition to displaying the results of simulations in graphical form, you can also export simulation results to output files with the following formats:

- Touchstone format (S-, Y-, or Z-parameters) for circuit and EM simulations, using the NPORTF measurement. (When writing Y- and Z-parameters using output files, the y-matrix is multiplied by the reference impedance and the z-matrix is divided by the reference impedance.) If sweeps are being performed the format can also be MDIF format. See “Generate Touchstone, MDIF, or MATLAB File: NPORTF” for details.
- AM to AM, AM to PM, or AM to AM/PM files for nonlinear circuit simulations, using the AMtoAMPMF measurement. See “Generate AM to AM/PM at Fundamental: AMtoAMPMF” for details.
- Spectrum data files for nonlinear circuit simulations, using the PharmF measurement. See “Generate Spectrum File: PharmF” for details.
- MATLAB "MAT" data files, using the MATLAB measurement. See “Write Measurement Data to MATLAB File: MATLAB” for details.

- Radiation pattern data files, using the AntPat_EF or AntPat_TPwrF measurements. See “Write Total Power Radiation Pattern to File: AntPat_TPwrF” and “Write E-Field Radiation Pattern to File: AntPat_EF” for details.
- NETDMP can generate a non-simulation based netlist of the circuit. See “Generate Netlist: NETDMP” for details.
- SpiceF can generate an RLC approximation of a passive circuit. See “Generate Spice Netlist Equivalent: SpiceF” for details.

NOTE: You must simulate the project after adding an output file to generate the file, regardless of whether previous simulations were run or not.

If the file format can also be used in the AWR Design Environment platform for other purposes, there is an option to import the file after simulation. Each subsequent simulation overwrites the data file in the project. To keep a permanent copy, rename the data file.

7.3.1. Creating an Output File

To create an output file to store the results of a simulation:

1. Choose **Project > Add Output File** or right-click the **Output Files** node in the Project Browser and choose **Add Output File**. The Add Output File dialog box displays. See “Add/Modify Output File Dialog Box” for more information about this dialog box.
2. Choose the measurement that corresponds to the type of data file you want to create. Click the **Meas Help** button for details on each measurement.
3. Specify the **Data Source Name**, the **File Name**, and the required options, and then click **OK**.

Chapter 8. Data Reports

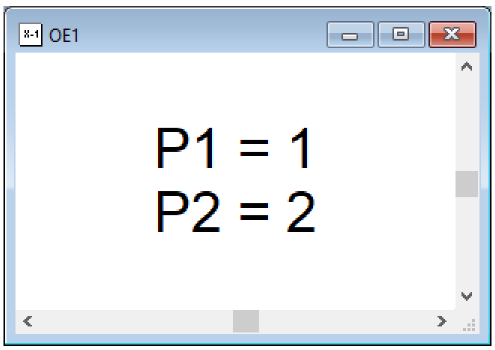
The Cadence® AWR Design Environment® platform allows you to combine measurement variables, Document Sets, and Window-in-windows together in an Output Equation document to create data reports with graphs and embedded windows that automatically update when measurement parameters and/or data sources are changed.

8.1. Measurement Variables

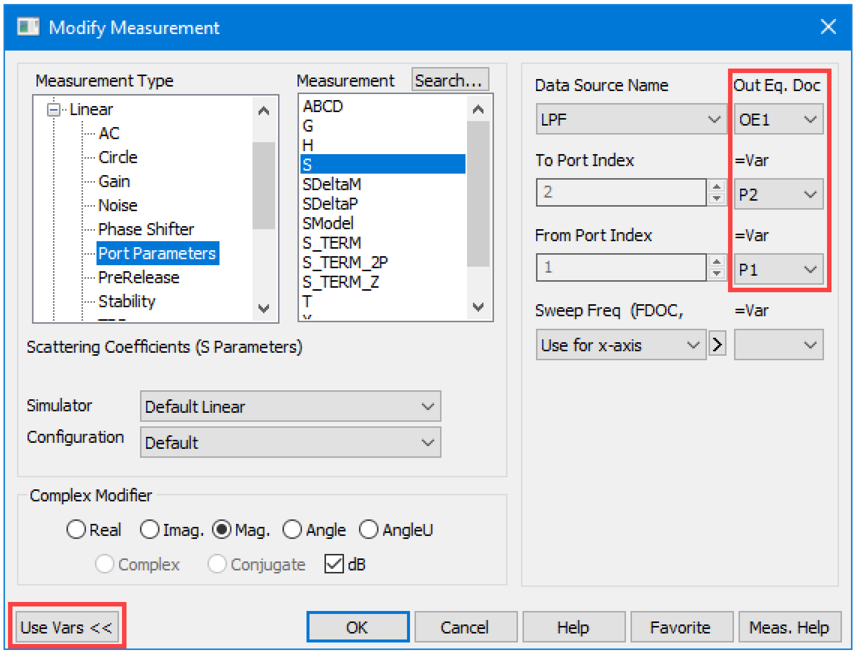
Variables defined in an Output Equation document can be used as a measurement parameter. See "*Variables and Equations*" for information on how to add and edit variables. A measurement variable does not need to be an independent variable, but is subject to these limitations:

- A measurement variable must be scalar.
- A measurement variable cannot depend on a measurement.

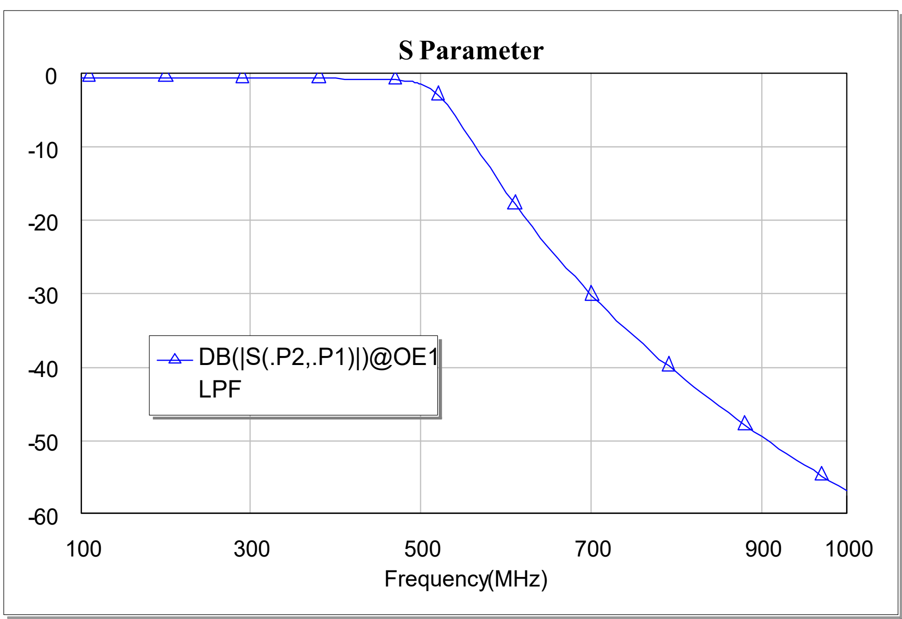
The following example shows how to use measurement variables. First define the variables in an Output Equation document, preferably with a short name.



In the Add/Modify Measurement dialog box, click the **Use Vars** button to expand the dialog box. In the **Out Eq. Doc** drop-down list, select the Output Equation document that is used to define the measurement variables, then select the measurement variable in the drop-down list next to the corresponding measurement parameter. You can widen this dialog box to display long Output Equation document or variable names.



Note that the measurement string in the following graph includes the Output Equation document and variable names. The measurement variables in the previous figure are set up to plot S21. By changing the values for the variables "P1" and "P2", you can easily plot different elements of the S-parameter matrix without adding new measurements. You can also enable tuning on "P1" and "P2", and use the tuner to quickly update the graph.



8.1.1. Supported Measurement Parameter Control Types

You can use measurement variables with the following control types:

Control Type	Variable Type	Example Measurement Parameter	Notes
Spinner	Integer	**To/From Port Index** in Port Parameter measurements	Port index range always starts at 1.
Integer	Integer	**Number of circles** in Circle measurements	
Real	Real	**Z0, real/imaginary** in Load Pull measurements	
Check Box	Integer	**Include Losses** in Antenna measurements	A value of "0" represents an unchecked box, and any other integer value represents a checked box.
Enumerated List	Integer	**Window Type** in TDR measurements	The value corresponds to the index in the options list. A value of "1" represents the first entry in the list. The integer value must be within the range of the number of entries in the drop-down list.
Swept Parameter (Variable, Frequency, Power, Voltage/Current)	Integer	**SWPVAR** in measurements on swept data sources	The value corresponds to the index in the swept parameter list. A value of "1" represents the first sweep point in the list.

8.1.2. Measurement Limitations

Measurement variables are only supported for measurements made on graphs and output equation documents. Annotations do not support measurement variables. Also, not all measurement parameters support measurement variables. The **Var** drop-down list does not include parameters that do not support measurement variables.

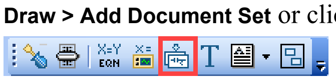
8.2. Document Sets

A Document Set represents a group of simulation documents. Adding a measurement on a Document Set is equivalent to adding measurements on all of the individual documents inside the Document Set. A Document Sets is defined either by a DOC_SET element in an Output Equation document, or by a User folder set up as a data source group.

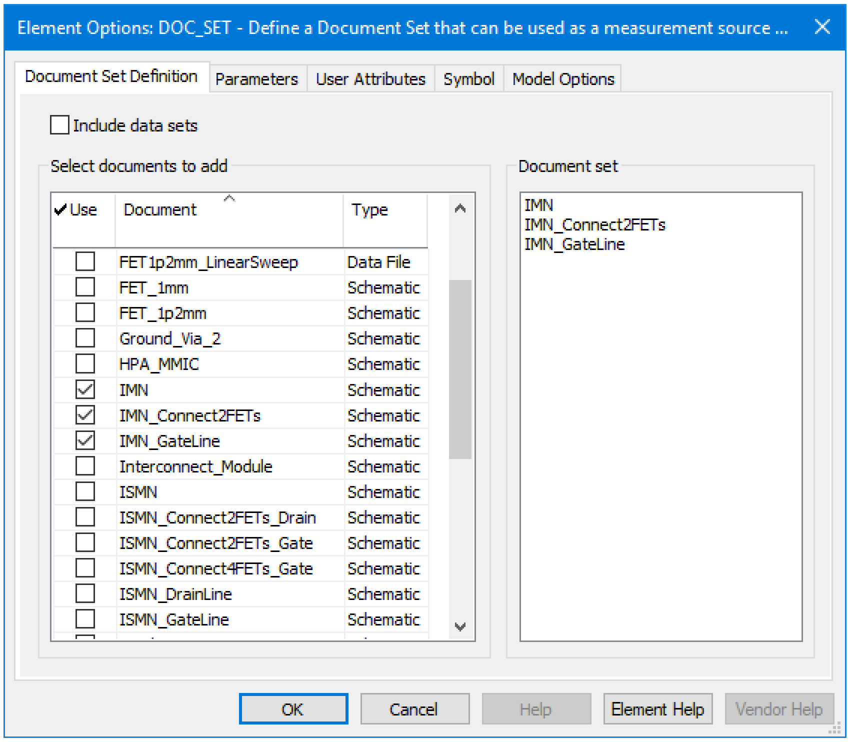
8.2.1. Working with DOC_SETs

8.2.1.1. Adding a New DOC_SET

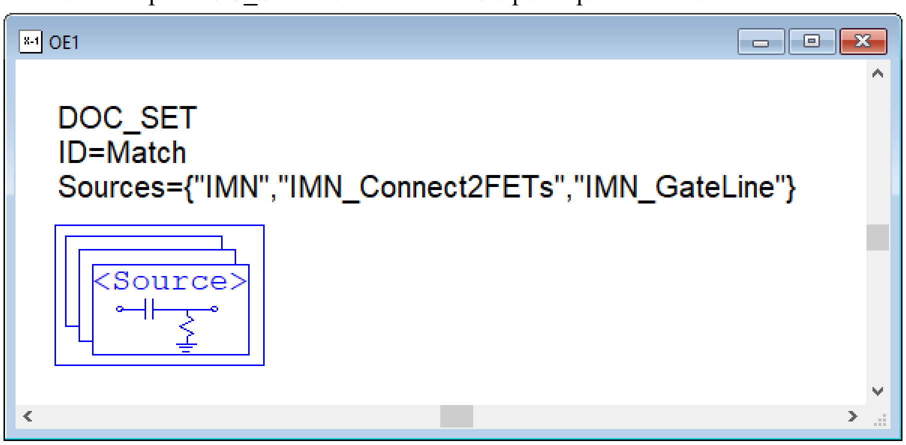
To create a Document Set using a DOC_SET element, add a DOC_SET to an Output Equation document by choosing **Draw > Add Document Set** or clicking the **Document Set** button in the Output Equations toolbar.



Click in the Output Equation document to add the DOC_SET element. The Element Options dialog box displays with a list of available data source documents in the project. Select the check box to include the document in the Document Set. The selected documents are added to the **Document Set** list.

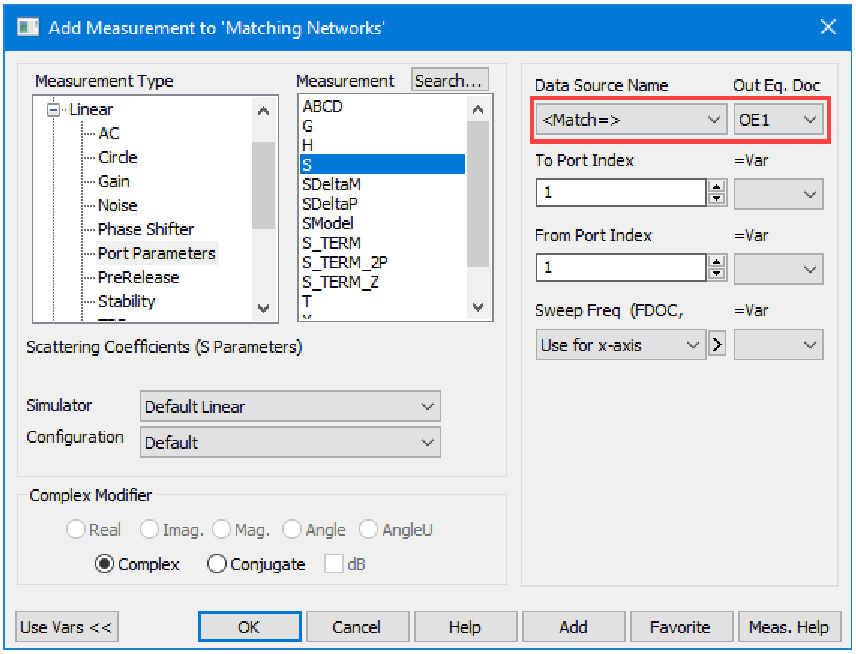


The selected documents also display as a list in the DOC_SET Sources parameter in the Output Equation window. You can add multiple DOC_SET elements in an Output Equation document.

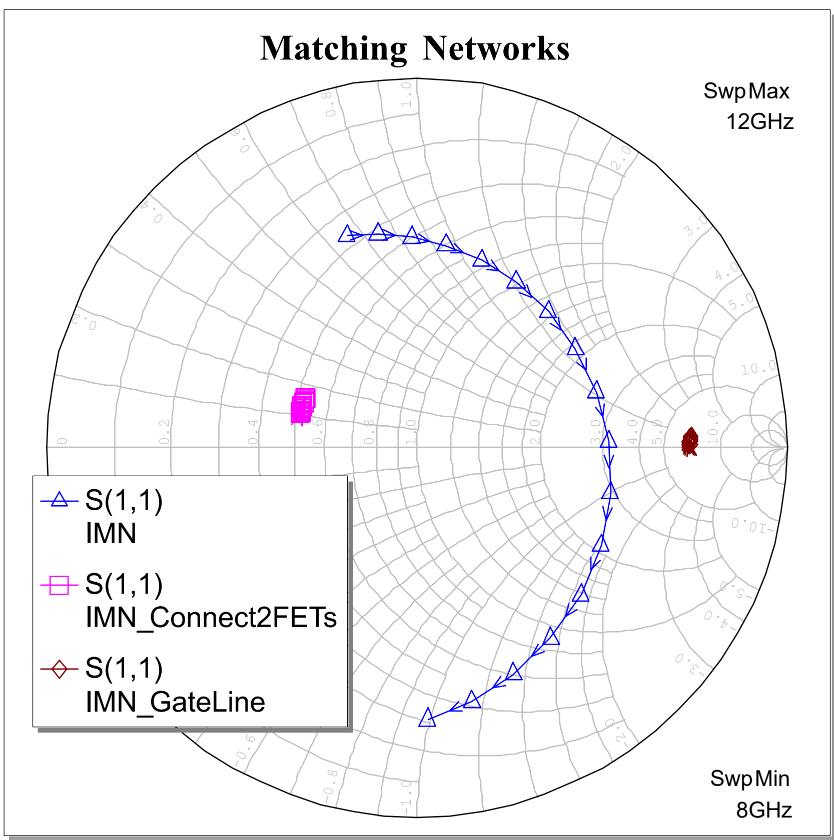


8.2.1.2. Using a DOC_SET in a Measurement

To make a measurement on a DOC_SET, click the **Use Vars** button in the Add/Modify Measurement dialog box to expose additional parameters. In the **Out Eq. Doc** drop-down list, select the Output Equation document that is used to define the DOC_SET element. In the **Data Source Name** drop-down list, select the Document Set defined by the <DOC_SET ID=> naming convention.



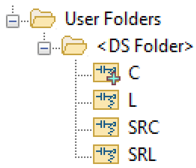
A measurement on the DOC_SET displays on a graph as individual measurements on each document in the Document Set. You can change the documents plotted on the graph by changing the selected documents in the DOC_SET, without needing to edit the measurement.



NOTE: When a DOC_SET is used as the data source for an output equation, only the first document (in alphabetic order) is used for the output equation. The variable assigned to the output equation can not support multiple measurements.

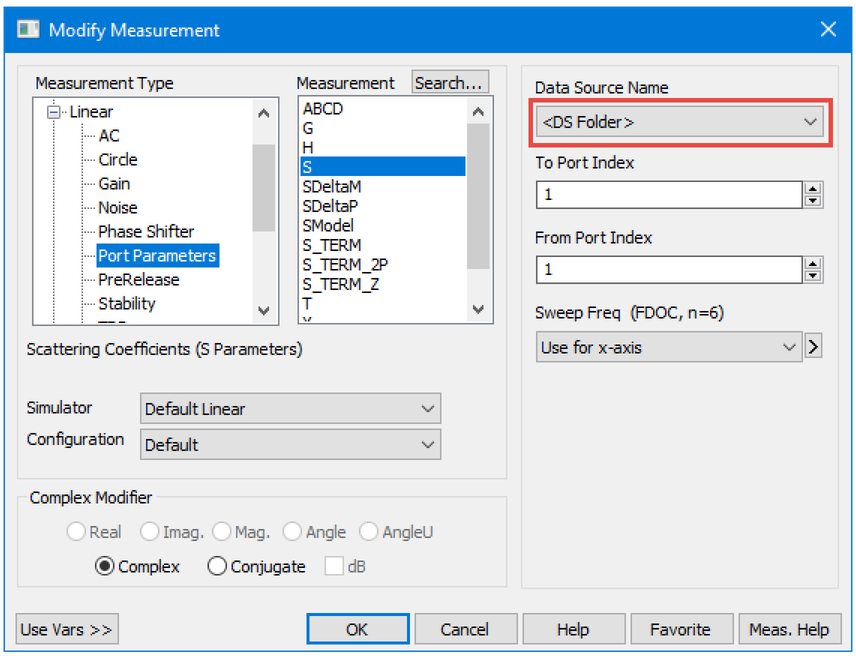
8.2.2. Working with Data Source Groups

You can also define a Document Set using a User folder set up with the data source group name convention. See "Grouping Collections Networks as a Document Set" for details on how to add a data source group.



8.2.2.1. Measurement on All Documents

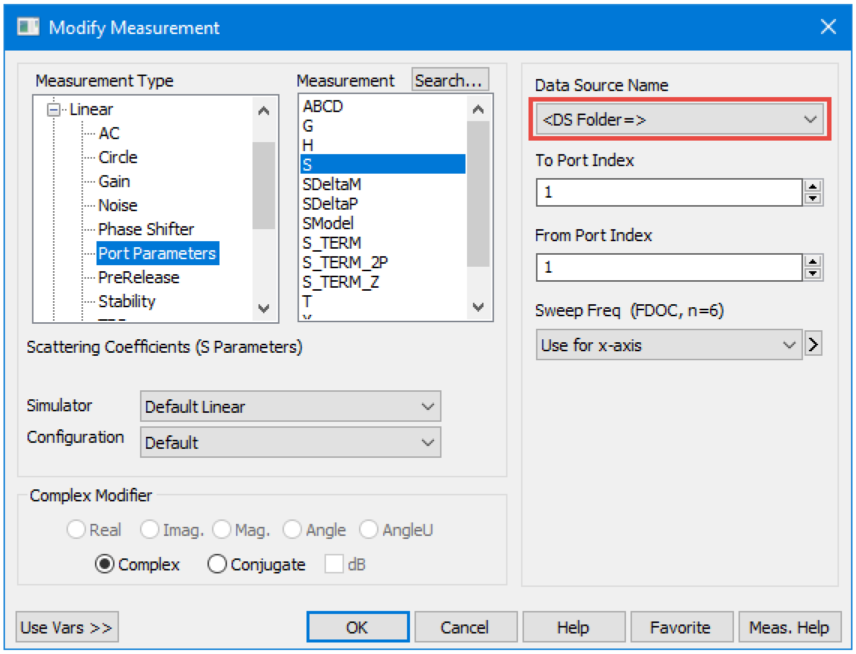
To make a measurement on all documents in a data source group, select <Folder Name> as the **Data Source Name** in the Add/Modify Measurement dialog box.



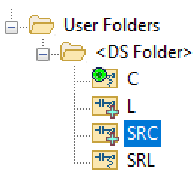
The measurement on the data source group expands as individual measurements on a graph for each document in the folder. The individual measurements automatically update as documents are added or removed from the data source group.

8.2.2.2. Measurement on Pinned and Active Documents

Data source groups also support another mode in which only selected documents are included in the Document Set. In the Add/Modify Measurement dialog box, choose <Folder Name => as the **Data Source Name**. Note the addition of the "=" in the naming convention.



Under the data source group node in the Project Browser, select a document to activate and include it in the Document Set. Selecting another document deactivates the previously selected document. **Ctrl**-click to select and activate multiple documents. A "+" sign displays on the icon of the active documents. You can also pin documents, so that the pinned document always remains included in the Document Set. To pin a document, right-click it and choose **Pin active document**. A green circle displays on the icon of pinned documents. To remove a pinned document from the Document Set, right-click it and choose **Unpin active document**.

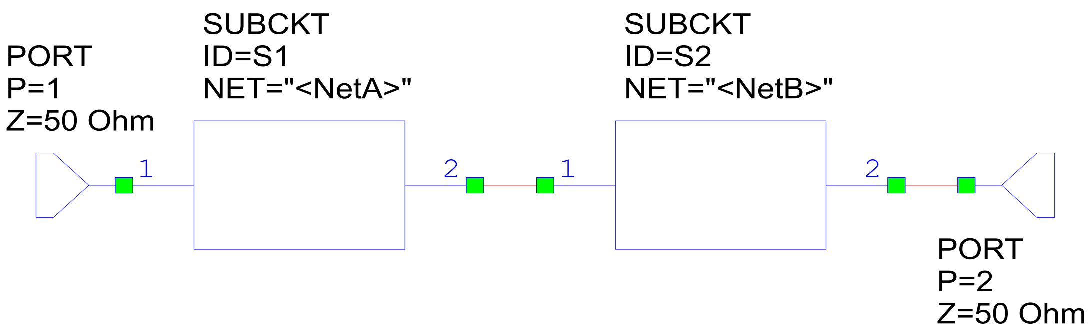


In this mode, measurements on the graph are only generated for the pinned or active documents. The graph automatically updates the measurement results as you click to change document selection.

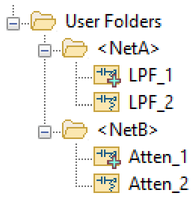
8.2.2.3. Template Documents

You can also use a data source group as a subcircuit template document by changing the SUBCKT NET parameter to the data source group name using the <Folder Name> convention. When you make a measurement on a schematic containing template documents, you can choose to plot the measurement results of any or all the permutations of the networks in the data source group collections.

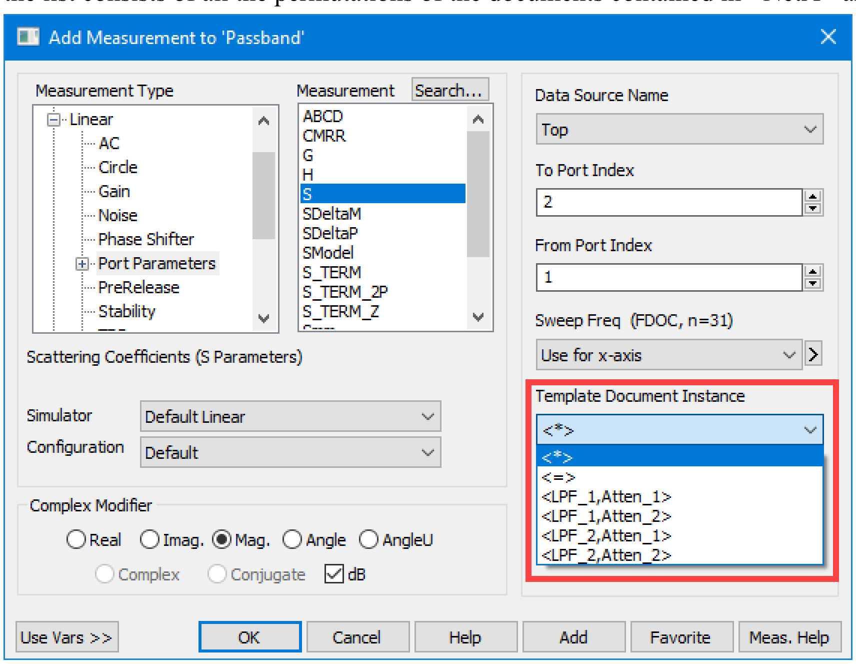
The following figure shows a schematic named "Top" that contains two template document subcircuits, <NetA> and <NetB>.



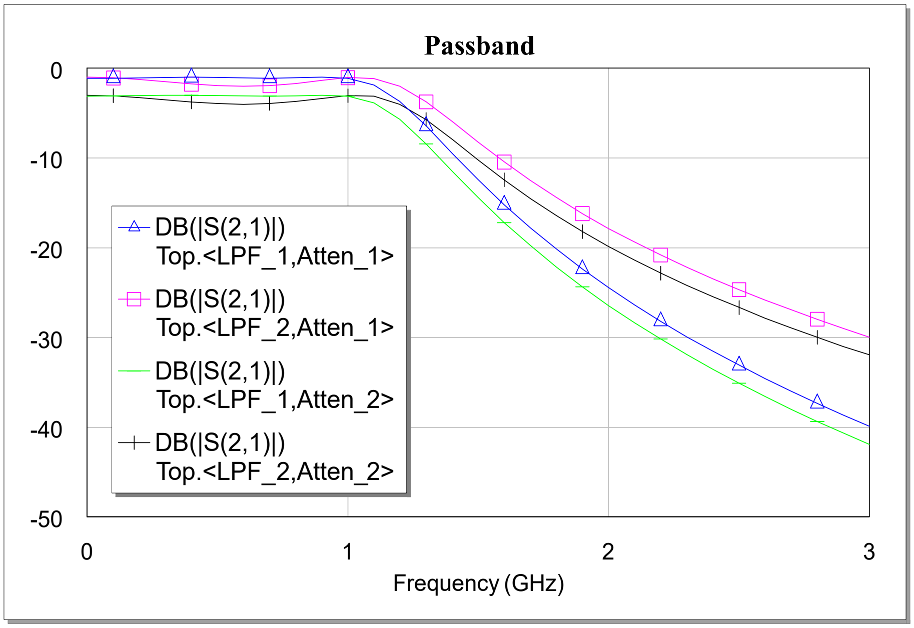
Both <NetA> and <NetB> contain multiple documents.



The Add/Modify Measurement dialog box contains an additional **Template Document Instance** drop-down list for measurements on schematics containing template documents. Choose <*> to plot results for all permutations of the documents in <NetA> and <NetB>. Choose <=> to plot the results of the active documents, which displays a "+" on its icon. See "Measurement on Pinned and Active Documents" for details on how to activate documents. The remainder of the list consists of all the permutations of the documents contained in <NetA> and <NetB>.

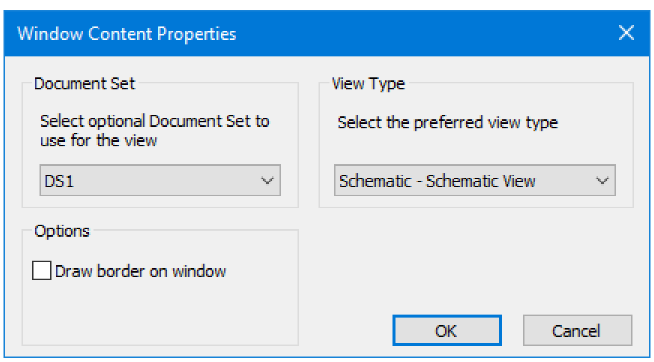


The following graph shows the results when you select <*> for the **Template Document Instance**. The measurement expands as individual measurements on a graph for all permutations of the networks in the data source group collections with the naming convention for each measurement as TopSchematicName.<InstanceX, InstanceY, ...>.



8.2.3. Synchronizing Window-in-window

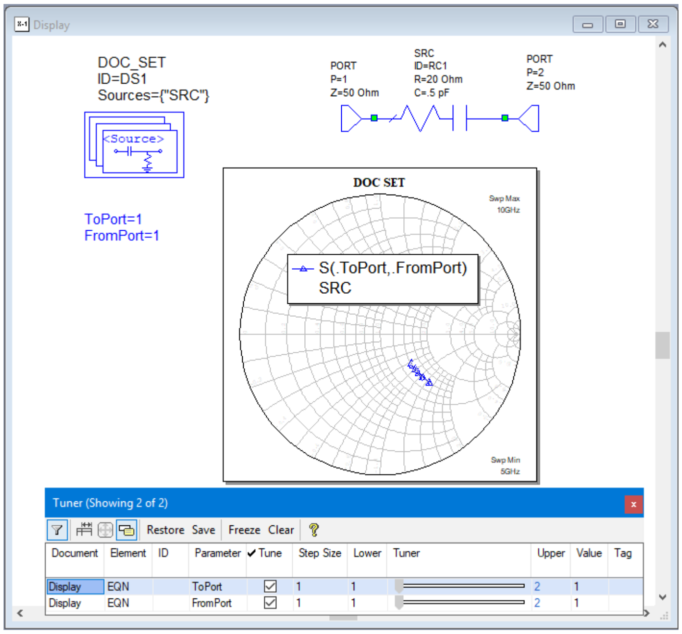
A Window-in-window object created in an Output Equation document can be synchronized to match a document in a Document Set. To associate a Window-in-window object with a Document Set, right-click the Window-in-window object and choose **Properties**. In the Window Content Properties dialog box select the name of the Document Set for synchronization, and the **View Type**, since documents can support multiple views. The Document Set can be either a DOC_SET element in the same Output Equations document as the Window-in-window object, or a data source group folder.



If there are multiple documents in the Document Set, the Window-in-window view is synchronized to the first document in the DOC_SET sources list, or first document in the data source group folder.

8.3. Working with Data Reports

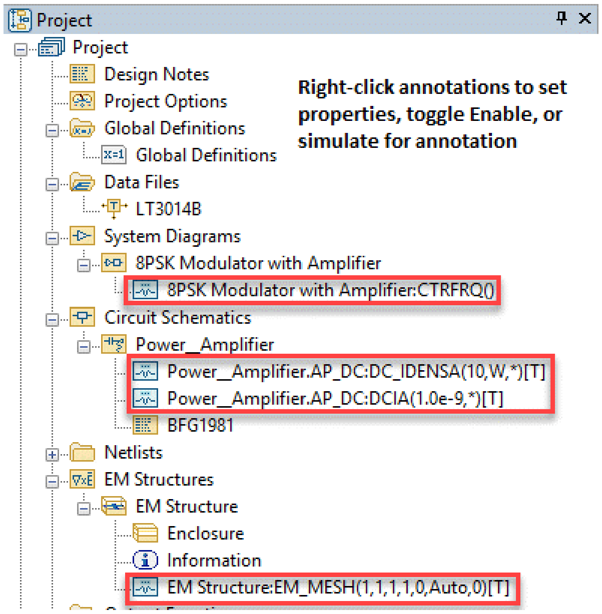
The following figure shows how to combine measurement variables, Document Sets, and Window-in-windows objects to create a data report in an Output Equation document. The Document Set in this example is defined using a DOC_SET element, but the concepts apply to data source folders as well. The S-parameter measurement in the Window-in-window graph uses the DOC_SET and measurement variables defined in the Output Equations document. You can change or add more measurement data sources by modifying the DOC_SET sources list, and you can tune on the variables to plot different port indices. The Window-in-window schematic view is paired with the DOC_SET, so the schematic document displayed changes to match the DOC_SET source.



Chapter 9. Annotations

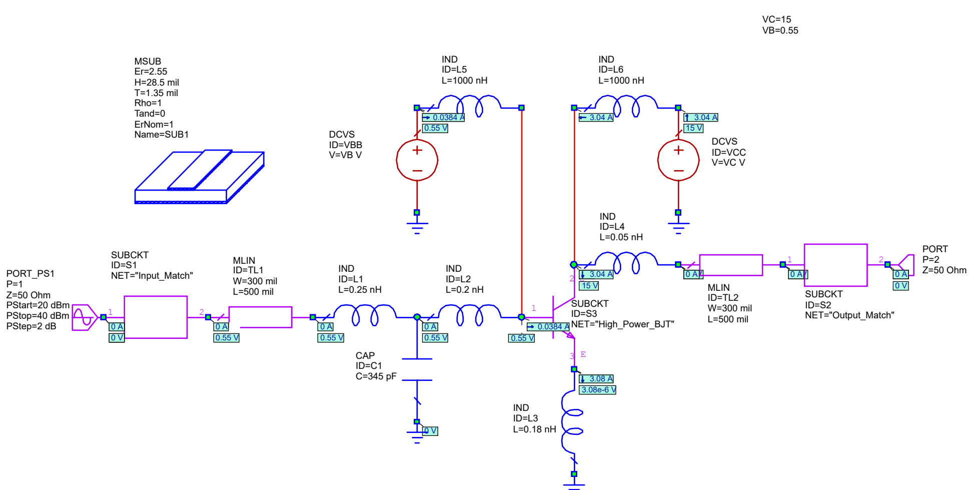
Annotations are simulation results plotted directly on schematics, system diagrams, or EM structures. Common examples are the DC current and voltage at each node for schematics, the center frequency for system diagrams, and the mesh for EM structures.

Annotations display under the **Circuit Schematics**, **System Diagrams**, and **EM Structures** nodes in the Project Browser when you add an annotation to these documents.



9.1. Working with Annotations

Annotations display directly on a schematic, system diagram, or EM structure. For example, the following figure shows where DC current and voltage are annotated on a schematic.

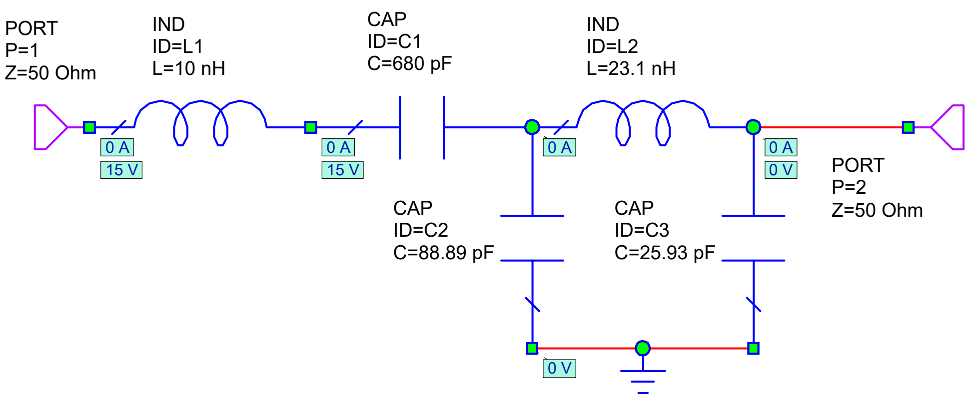


NOTE: For two-port elements, current annotation always displays on node 1.

See "EM Annotations and Cut Planes" for more details on EM annotations.

9.1.1. Hierarchy

Schematics and system diagrams are commonly created using hierarchy. When an annotation is applied to a top level schematic or system diagram, you can push down through the hierarchy to see the annotations. To do this, select any subcircuit in the top level, right-click and choose **Edit Subcircuit**. You descend into that subcircuit and can see the annotations from the top level displayed at the lower level. From the previous example, see the following figure of the annotation in the "Output Match" subcircuit.



If you open the subcircuit from the Project Browser, there is no annotation display because the Cadence® AWR Design Environment® platform does not know every place this subcircuit is used at a higher level.

9.1.2. Creating a New Annotation

You can create a new annotation by right-clicking the following nodes in the Project Browser and choosing **Add Annotation**:

- a circuit schematic node under **Circuit Schematics**
- a system diagram node under **System Diagrams**
- an EM structure node under **EM Structures**

or you can select these nodes and click the **Annotation** button on the toolbar.

An Add Schematic Annotation, Add System Diagram Annotation, or Add EM Structure Annotation dialog box displays, depending on the node. See "Add/Edit Schematic/System Diagram/EM Structure Annotation Dialog Box " for more information.

Select the **Measurement Type** and the **Measurement** to add the annotation, then click **OK**. The Project Browser displays the new annotation under the appropriate node for the item.

Annotations function identical to graphs in regards to tuning and swept parameters. See "Swept Parameter Analysis " for more information on swept parameter analysis and results display.

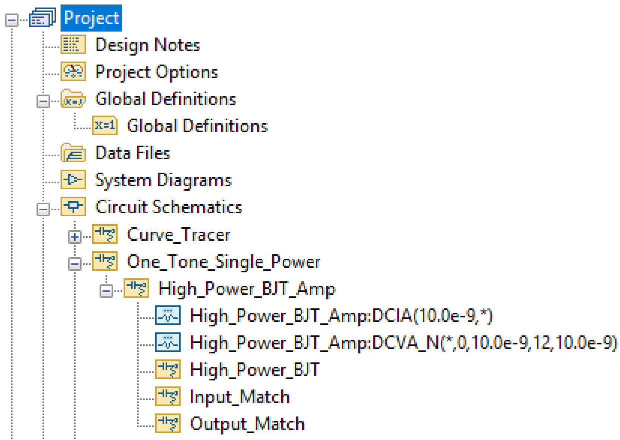
For more information about adding back annotations, see "Adding Back Annotation to a Schematic or System Diagram".

9.1.3. Modifying the Annotations Display

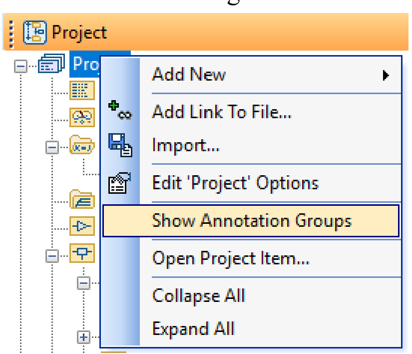
You can control how annotations display in the Project Browser and on schematics or system diagrams. Choose **Options > Environment Options** to display the Environment Options dialog box, then click the **Schematic Annotation** tab to edit annotation display properties on documents. See "Environment Options Dialog Box: Schematic Annotation Tab " for details on annotation settings.

9.1.3.1. Changing Annotations in the Project Browser

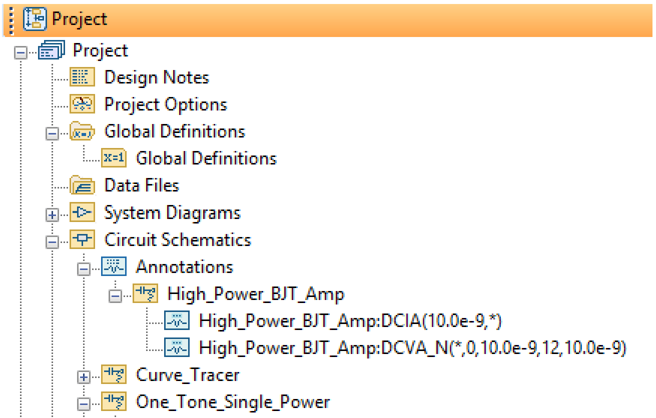
By default, in the Project Browser annotations display directly under each schematic, system diagram, or EM structure, as shown under the "High Power BJT Amp" schematic in the following figure.



You can display them under separate **Annotations** subnodes if you prefer, by right-clicking the **Project** node of the Project Browser and choosing **Show Annotation Groups**.



For example, see the **Annotations** node with two schematic annotations now included in the following figure.

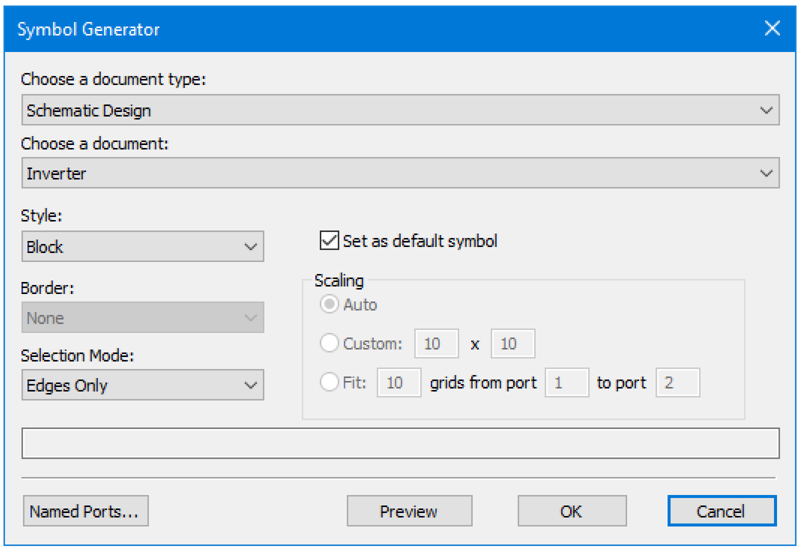

Project
Project
Design Notes
Project Options
Global Definitions
Global Definitions
Data Files
System Diagrams
Circuit Schematics
Annotations
High_Power_BJT_Amp
High_Power_BJT_Amp:DCIA(10.0e-9,*)
High_Power_BJT_Amp:DCVA_N(*,0,10.0e-9,12,10.0e-9)
Curve_Tracer
One_Tone_Single_Power

Chapter 10. Circuit Symbols

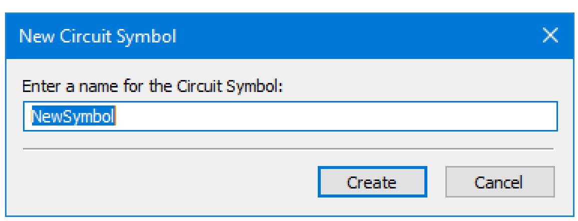
The Cadence® AWR Design Environment® platform allows you to create and use your own symbols for any model, including subcircuits. The Symbol Generator Wizard can draw symbols to match the shapes of a layout. The Symbol Editor allows you to create, rename, and edit symbols, and to export them to a symbol (*.syf*) file. You can also import existing symbols into your projects, link to symbol files, and edit both imported and linked files.

10.1. Adding Symbols

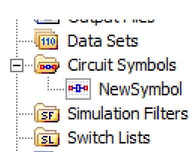
The Symbol Generator Wizard allows you to create and edit custom symbols for a schematic or EM document and save them with the current project. After choosing an active schematic you can select one of three styles from which to create the symbol: schematic, layout, or block style. To access the Symbol Wizard open the **Wizards** node in the Project Browser. For more information about using this wizard, see "Symbol Generator Wizard".



To create a new symbol without using the wizard, right-click the **Circuit Symbols** node in the Project Browser and choose **New Circuit Symbol**. Alternatively, choose **Project > Circuit Symbols > Add Symbol**. A New Circuit Symbol dialog box displays.



Type a name for the new symbol and click the **Create** button to display a window (the Symbol Editor) with a default symbol to edit. The symbol displays as a node under **Circuit Symbols** in the Project Browser.



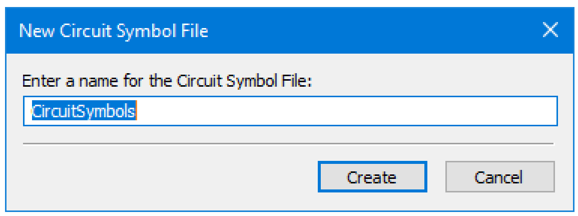
10.1.1. Importing Symbols

To import symbols for use in the current project, right-click the **Circuit Symbols** node and choose **Import Circuit Symbols**. Alternatively, choose **Project > Circuit Symbols > Import Symbols**. In the “Import Symbols Dialog Box ”, browse to the *.syf* file containing the symbols you want to import (the *\symbols* subdirectory of the program installation directory contains the symbols provided with the software) and select the desired symbol(s).

10.1.2. Linking to Symbol Files

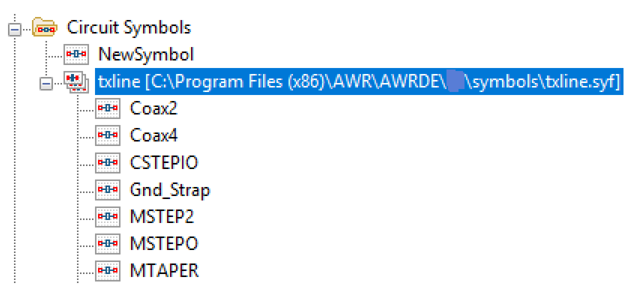
You can create a linked symbol file or link to an existing symbol file and use the symbols in that file in the current project. If the linked file is modified externally, its symbols are automatically updated within the AWR Design Environment platform project. Changes to the symbols while the project is open are saved into the linked symbol file when you save the project. If the symbol file is read-only, you cannot edit the linked symbols in the Symbol Editor and "(file locked)" displays in the window title bar.

To create a linked symbol file, right-click the **Circuit Symbols** node in the Project Browser and choose **New Circuit Symbol File**. A New Circuit Symbol File dialog box displays.



Type a name for the new symbol file and click the **Create** button to add a linked circuit symbol file node beneath the **Circuit Symbols** node. The file path displays in brackets and the node icon depicts multiple symbols. You can add symbols to the file by right-clicking the node and choosing **New Circuit Symbol**.

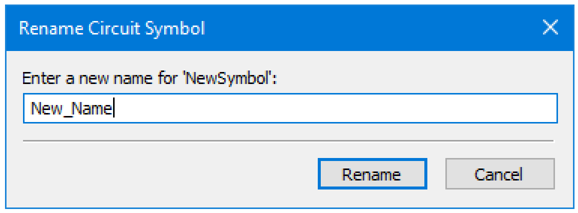
To link to an existing symbol file, right-click the **Circuit Symbols** node in the Project Browser and choose **Link to Circuit Symbol File**. In the Link to Symbols File dialog box, navigate to the symbols file (*.syf*) you want to link to and click **Open**. The file path displays in brackets beneath the **Circuit Symbols** node, with the symbols it contains indented underneath. The node icon depicts multiple symbols.



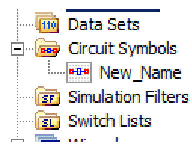
Right-click on the name of linked symbol files for the following additional commands: **Delete**, **Embed**, **Reload**, **Save**, **Save As**, and **Explore**. The **Embed** command moves all of the symbols listed under the file into the top level **Circuit Symbols** folder, embedding them in the project instead of linking to them. The **Explore** command opens the directory in which the linked file is stored.

10.2. Renaming Symbols

To rename a symbol, right-click the symbol node in the Project Browser and choose **Rename**. Alternatively, choose **Project > Circuit Symbols > Manage Symbols** to display the Manage Symbols dialog box. Select the symbol you want to rename and click the **Rename** button to display the Rename Circuit Symbol dialog box.



Type a new symbol name and click the **Rename** button to save the change. The symbol displays the new name in the Project Browser.



10.3. Deleting Symbols

To delete a symbol, right-click the symbol in the Project Browser and choose **Delete**. You can also select the symbol in the Project Browser and press the **Delete** key. Press **Shift + Delete** to delete without a prompt to confirm the deletion. Alternatively, choose **Project > Circuit Symbols > Manage Symbols** to display the Manage Symbols dialog box. Select the symbol you want to delete and click the **Delete** button.

You can automatically remove symbols that are not currently in use by right-clicking the **Circuit Symbols** node in the Project Browser and choosing **Delete Unused Circuit Symbols**.

10.4. Copying Symbols

There are several ways to duplicate symbols:

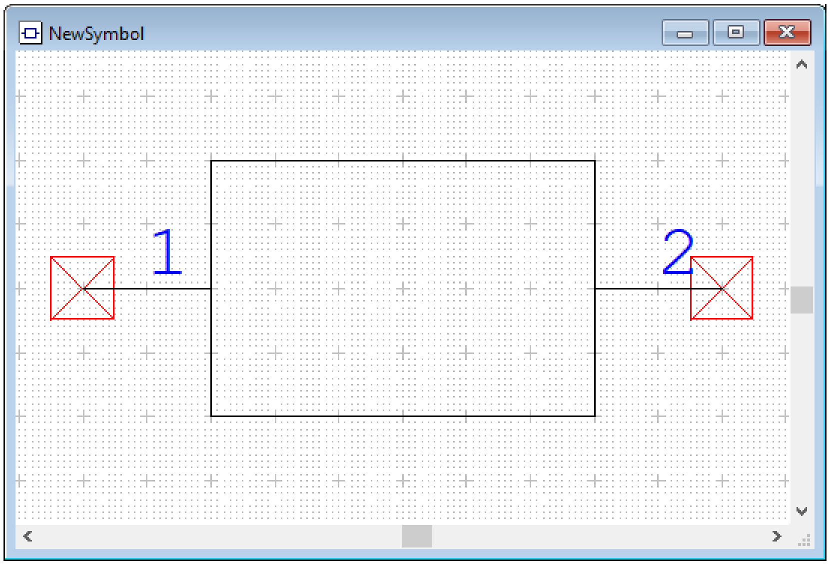
- Right-click the symbol in the Project Browser and choose **Duplicate**.
- Drag and drop the symbol in the Project Browser onto the **Circuit Symbols** node.
- Select the symbol in the Project Browser and press **Ctrl+C** and then **Ctrl+V**.

10.5. Exporting Symbols

To export symbols for use in another project, right-click the **Circuit Symbols** node and choose **Export Circuit Symbols**. Alternatively, choose **Project > Circuit Symbols > Export Symbols**. In the "Export Symbols Dialog Box ", specify the name of a new symbol file to contain the exported project symbols.

10.6. Using the Symbol Editor

To edit an existing symbol, double-click the symbol name in the Project Browser. Alternatively, select the symbol name in the Manage Symbols dialog box and then click the **Edit** button to open the symbol in the Symbol Editor window.



The Symbol Editor uses an integer drawing system. The smallest you can draw a point is 1 grid point. The minor grid (displayed as the smaller dots in the Editor) are 10 grids across. The major grid (displayed as the + sign in the Editor) are 100 grids across. Choose **Draw > Grid Snap** to allow new items added to the symbol to snap to the minor grid.

The Symbol Edit toolbar contains buttons for adding nodes, drawing various shapes, and adding text. The same commands are available on the **Draw** menu.

NOTE: When adding vertices for shapes in the Editor, you can use coordinate entry just as you can when using the Layout Editor. See "Coordinate Entry" for more information on coordinate entry.

10.6.1. Adding Nodes

When adding a node, a ghost image of the node follows the cursor location. You can only place nodes on the major grid locations (multiples of 100). Click to add the node to the current location. Nodes are assigned the next available number. After you place the node, you can select the node text to move it to a new location and double-click the text to change it. With the node text selected, choose **Draw > Label Visible** to toggle the label display. When disabled, the text displays in gray, and when you use the symbol the text from the node does not display.

Cadence suggests the following for nodes:

- Nodes on opposite sides of the symbols should be 1000 grids (10 major grid points) apart. Standard symbols all use this spacing, which helps to maintain the schematic connectivity when switching a model's symbol.
- Do not put shapes or text outside of a node. This can make selecting other symbols difficult in the Symbol Editor.

You must follow these node rules:

- Node names can be numbers or strings. A single symbol can have numbers and strings for the node name.
- Numbered node names must start with 1 and must be sequential.
- Strings are used with PORT_NAME elements; the string must match the PORT_NAME exactly, including case and any vector instance syntax.

10.6.2. Adding Rectangles

When adding a rectangle, click to add the first point, then continue holding down the mouse button while dragging to create the rectangle. Release the button to finish adding.

10.6.3. Adding Polylines

When adding a polyline, click to add the first point, then move the cursor and click to add additional points. Double-click to finish adding the polyline.

10.6.4. Adding Ellipses

When adding an ellipse, click to add the first point, then continue holding down the mouse button while dragging to create the ellipse. Release the button to finish adding.

10.6.5. Adding Arcs

When adding an arc, click to add the first point, then continue holding down the mouse button while dragging to create the arc. Release the button to finish adding.

10.6.6. Adding Text

When adding text, a ghost image of the text follows the cursor location. Click to set the text location and then type the desired text. Press **Enter** to finish adding text.

10.6.7. Update Symbol Edits

Click the **Update Symbol Edits** button to save the current edits. You can also close the symbol window and click **Yes** when prompted to save the symbol.

10.6.8. Editing Symbol Shapes

After you add symbol shapes, you can edit them as follows:

- Select individual or groups of shapes to move them.
- Select groups of shapes and use the **Align** toolbar or choose **Draw > Align Shapes** or **Draw > Make Same Size** to align and resize shapes.
- Double-click a shape to edit either its vertices or text. Choose **Draw > Orthogonal** to allow shape edits to move in one direction only.
- Select individual or groups of shapes, then right-click and choose to **Flip** or **Rotate** the shapes. Nodes do not have these options.
- Select individual or groups of shapes and choose **Draw > Snap Shapes to Grid** to snap any vertices to the minor grid.

10.7. Using Symbols

Each model in the AWR Design Environment platform is assigned a default symbol, however you can change these symbols. Additionally, you can assign a default symbol to items that can be used as subcircuits (schematics, data files, and EM structures).

10.7.1. Changing Symbols

To change a model symbol:

1. Double-click the model in the schematic or system diagram to display the Element Options dialog box.
2. Click the **Symbol** tab. See "Element Options Dialog Box: Symbol Tab" for more information. The list of symbols is filtered so the number of nodes for the symbol match the nodes of the model. You can further filter the symbols by selecting the source of the symbol from the drop-down list. The default is all of the symbols in the project. Typically you should use the **Project** setting, which is for symbols created in the current project. A preview of the symbol displays in the dialog box.
3. Select the symbol you want and click **OK**.

10.7.2. Default Subcircuit Symbols

When you use a data file, EM structure, or schematic as a subcircuit in a schematic, the default symbol is a rectangle with the appropriate number of ports evenly distributed around each side. Using the steps from the previous section, you can change this symbol after the symbol is placed. Often you may want to use the same symbol, so for each type, you can assign the default symbol to use when it is a subcircuit. To assign a default symbol:

1. Right-click the specific document in the Project Browser and choose **Options** to display the Options dialog box.
2. Click the **Symbol** tab.
3. Select the desired symbol from the list of symbols and click **OK**.

10.7.3. Symbols in Library Elements

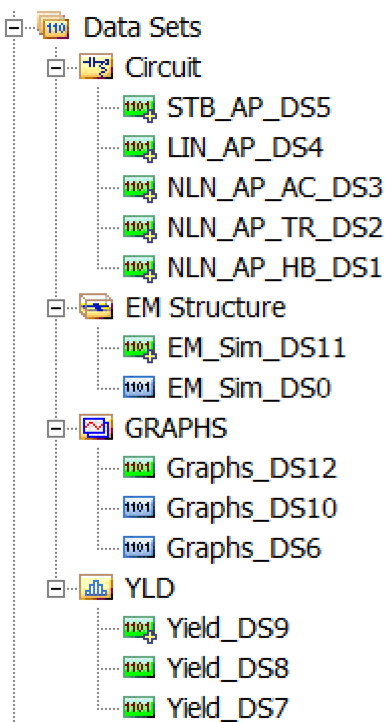
If you are building a library of elements, you can specify which symbol to use. See Appendix A, *Component Libraries* for more information.

Chapter 11. Data Sets

The Cadence® AWR Design Environment® platform supports three types of data sets:

- Graph data sets: storing/restoring data on graphs.
- Yield data sets: storing/restoring results from yield analysis.
- Simulation data sets: storing/restoring simulation results for a given type of simulation

The following figure shows the **Data Sets** node in the Project Browser with graph data sets under the **GRAPHS** and **YLD** nodes, and several simulation data sets under the **EM Structure** and **Circuit** (circuit simulation) nodes.



Graph data sets only store the current trace data that is on a graph when the data set is created. Restoring data from these data sets populates graphs with the data that existed when created. While these data sets restore the graph display, they do not make the simulators "clean", and you still need to simulate the project to make all data current. In addition, if you add measurements to a graph a new simulation is required to update the graph.

Yield data sets store the data collected when running yield analysis for any measurements, as well as the data needed to plot any other yield type information. Restoring data from these data sets populates graphs with the data that existed when created. While these data sets restore the graph display, they do not make the simulators "clean", and you still need to simulate the project to make all data current. In addition, if you add measurements to a graph a new yield analysis run is required to see the yield data on the new graph.

Simulation data sets store full simulation results from the simulator (data source), regardless of what is plotted on a graph. You can therefore add new measurements later using that data source. For example, if you run an EM simulation with Cadence AXIEM® 3D planar EM analysis software and only plot S(1,1) of the structure before simulation, you can then plot S(2,2) and the data set contains that data to plot without requiring a new simulation.

In general, all data sets are **.dsf* files stored on disk. There is an option to store them in the project. When stored on disk, by default they are created in the *DATA_SETS* folder in the project directory. You can set a different location for the **Data Set Directory** on the Environment Options dialog box **File Locations** tab. If you do not have write permission for the Data Set Directory folder, an error displays and simulation stops. **NOTE:** The AWR examples included with the software are an exception. They simulate correctly and the data sets are written to a temporary directory. This temporary directory has data removed every time the software starts, so if you want to keep data sets from standard examples you should save the example to another location.

11.1. Graph Data Sets

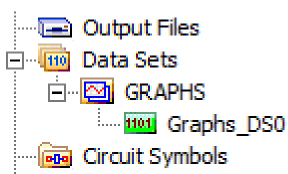
Graph data sets store the current data on all the graphs. Once a data set exists, you can restore the data from that data set to the graphs.

11.1.1. Adding Graph Data Sets

To create a graph data set:

1. Simulate your project so all data is current on your graphs.
2. Right-click the **Data Sets** node in the Project Browser and choose **Add Graph Results Data Set**.

   Under the **GRAPHS** node, a new data set displays, as shown in the following figure.



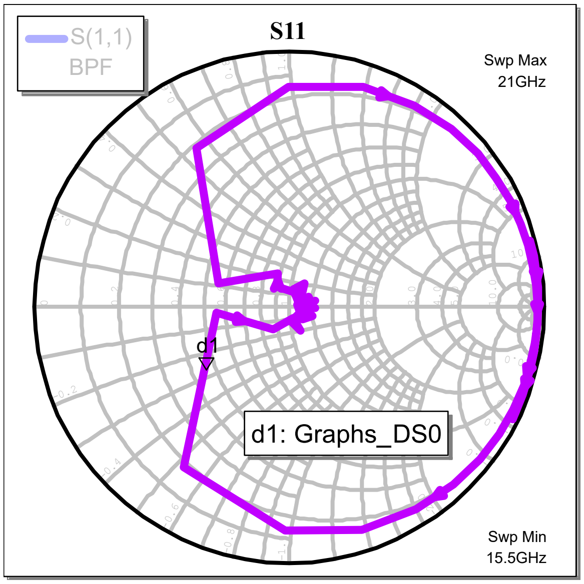
11.1.2. Restoring Data from Graph Data Sets

To update graphs to display data from a graph data set:

1. Right-click the data set of choice under the **GRAPHS** subnode of the **Data Sets** node in the Project Browser and choose **Show Results**, or press the **Shift** key and click on the data set.

   The data on the graphs now displays from the data set and not from the current simulation.
2. Simulate the project to clear the graph data set results and display the current simulation results.

As shown in the following figure, the data on a graph from a data set is different from current simulation results in the following ways:



- A data set marker on the graph shows the name of the data set being plotted. When you simulate to make the data current, these markers no longer display.
- The trace color is different, as specified in the graph trace properties. Each graph has property settings for controlling how to display different types of traces (frozen, yield and data sets). You can configure default graph options by choosing **Options > Default Graph Options > <graph type>** and clicking the **Format** tab, or you can set the properties for individual graphs by right-clicking a graph window and choosing **Properties** to display a dialog box with properties for that graph. See "Graph Options Dialog Box: Format Tab" for details.
- The graph legend is grayed out.

11.1.3. Automatically Saving and Restoring Graph Data Sets

You can configure data sets to automatically save a graph data set when you close or save a project, and restore the data to the graphs when you next open the project. You do not need to simulate to view your data when you open a project. To use this feature:

1. Right-click the **Data Sets** node in the Project Browser and choose **Options**.
2. In the Data Set Options dialog box, set **Auto Save Graph Data Set** to **On project close** to save any clean data when the project is closed, or **On project save** to save any clean data when the project is saved.
3. Select **Auto Restore Graph Data Set** to automatically restore graphs using the auto-saved graph data set when you reopen the project.

If the project is simulated, then edited, and then saved or closed, the automatically saved graph data set may not be considered clean or restored automatically. To manually restore any saved results, shift-click the graph data set, or right-click and choose **Show Results**. If one or more graphs that were present when the data set was saved are deleted before you save the project, you are prompted to recreate the graph(s). If a simulation result is not clean (there has been an edit to the circuit and no simulation performed since), the data is not saved to the graph data set and cannot be restored.

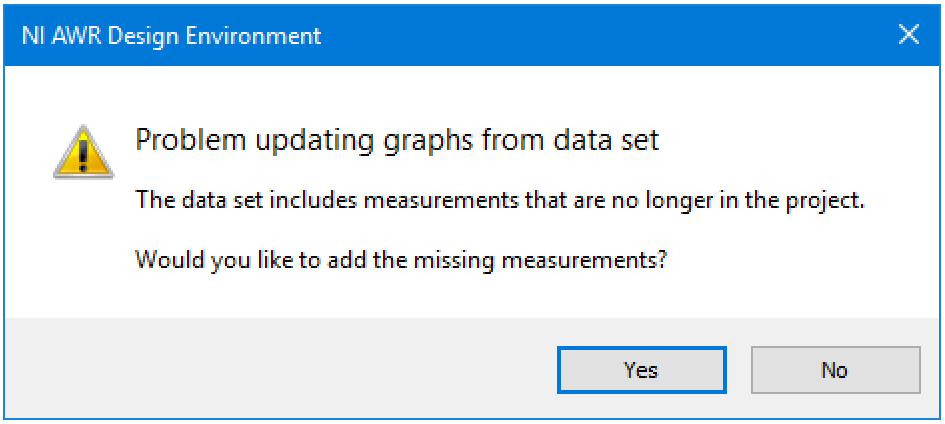
11.1.4. Using Graph Data Sets in a Blank Project

Graph data sets are also useful for viewing simulation results without needing to configure all of the simulations. To use a graph data set created from another project:

1. Import the data set by right-clicking the **Data Sets** node and choosing **Import Data Set**.

   The data set displays under the **GRAPHS** subnode of the **Data Sets** node in the Project Browser.

2. Right-click the imported data set and choose **Show Results** or press the **Shift** key and click on the data set. Because there are no graphs in the current project, you are prompted to create the graphs.



3. Click **Yes** to display the data on your graphs.

NOTE: Data sets do not store graph formatting information, so graphs are created with your default settings. If you have significant graph formatting you want to maintain, you have the following options.

1. For one or a few graphs, if you have both projects open, you can select a graph in the Project Browser and copy it (choose **Edit > Copy** or press **Ctrl+C**). In the new project, click the **Graphs** node in the Project Browser and paste the graph in the new project (choose **Edit > Paste** or press **Ctrl+V**).
2. For many graphs, where you can quickly select all graphs or just a subset to import, you can use the project import feature to import the graphs from the original project into your new project. See "Importing a Project " for details on how to use project import.

11.2. Yield Data Sets

Yield data sets store all the yield data from any simulator during yield analysis.

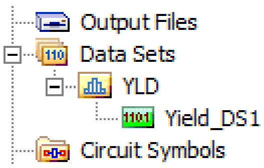
11.2.1. Adding Yield Data Sets

The simulation data set at each yield iteration is not saved by default during yield analysis. The default **No Yield Sims Retained** option allows simulation data sets to be saved. This option displays when you click the **Show Secondary** button in the Data Set Options dialog box. With this option selected, any simulator that supports data sets keeps a data set for each yield iteration when the yield analysis is done. The next simulation limits these data sets to the value specified in **Max Retained**.

You can select the **Create data set for yield analysis** check box in the Yield Analysis dialog box to automatically create a yield data set for the yield analysis run.

Alternatively, to add a yield data set after the yield runs:

1. Right-click the **Data Sets** node in the Project Browser and choose **Options**.
2. In the Data Set Options dialog box, select **Auto Save for Yield** and click **OK**.
3. Run a yield analysis and a new data set is automatically added under the **YLD** node.



Any measurements made from the **Yield** measurement category after a yield run is done will plot directly-- a new yield run is not necessary.

11.2.2. Restoring Data from Yield Data Sets

To update graphs to display data from a yield data set:

1. Right-click the data set of choice under the **YLD** subnode of the **Data Sets** node in the Project Browser and choose **Show Results**, or press the **Shift** key and click on the data set.

   The data on the graphs now displays from the data set and not from the current simulation.
2. Simulate the project to clear the graph data set results and display the current simulation results.

NOTE: You can also use yield data sets in blank projects, the same way you use graph data sets. See "Using Graph Data Sets in a Blank Project" for details.

11.3. Simulation Data Sets

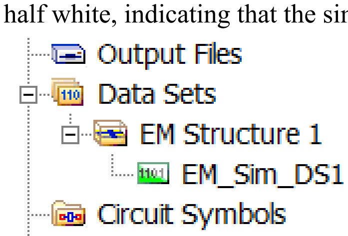
The AXIEM, Cadence Analyst™ 3D FEM EM, Cadence APLAC® HB, and Cadence Visual System Simulator™ (VSS) communications and radar systems design software simulators use data sets to store simulation data. Circuit schematics, system diagrams, and EM structures are all referenced as "source documents". When you run these simulations, the name of the source document displays under the **Data Sets** node, and the data sets for each simulation of that document are located under this node.

You can view the results from data sets as follows:

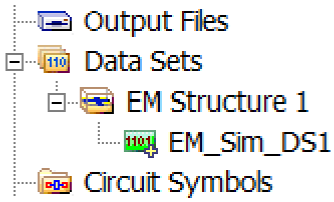
- If you want measurements to plot results from a specific data set and not be affected by future simulation, you create measurements that point directly to the data set.
- You can import data sets into different projects and plot data from those data sets. In this case, there is no source document to simulate; you are only viewing data from previous simulations.
- If you want measurements that update with each simulation, but want to be able to review old simulation results, you can choose the **Update Results** command. See "Updating Data Sets" for more information.

11.3.1. Data Set Icon Colors

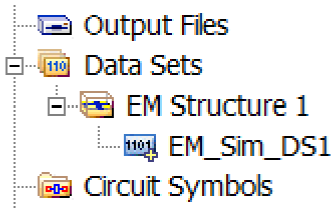
Each simulation data set icon has a meaningful color. When you start a simulation the icon displays in half green and half white, indicating that the simulation process is filling the data set.



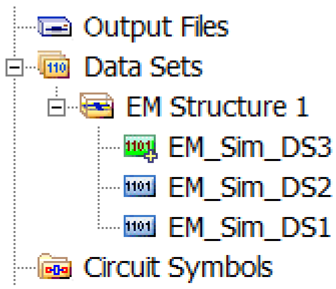
After the simulation is complete, the data set displays in solid green, indicating that it contains the data for the current state of the source document.



After you edit the source document, the data set displays in gray, indicating that it is an old data set.



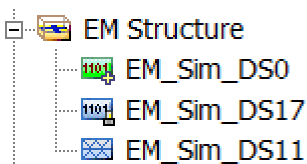
As you perform more simulations on the source document, you accumulate more data sets with the active data set (green) displaying on the top of the list.



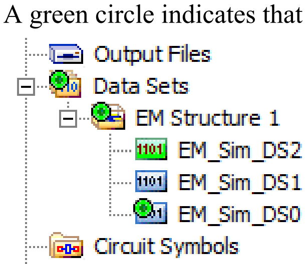
11.3.1.1. Data Set Icon Symbols

In addition to colors, data set icons can display the following meaningful symbols:

- A "+" sign indicates that the data set is being used by a simulator in the project. For example, the associated EM structure may be used as a subcircuit in a schematic.
- A lock indicates that the data set cannot be auto-deleted. See "Disabling Auto Delete" for more information.



- A green circle indicates that a data set is pinned. See "Pinning Data Sets" for more information.

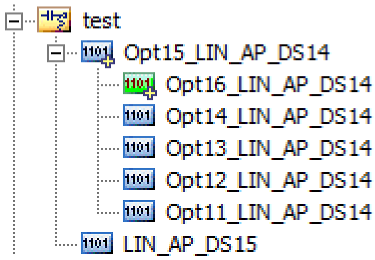


11.3.2. Data Set Accumulation

For each new simulation of a data source, a new data set is created. You can set the maximum number of data sets retained per source document by right-clicking the **Data Sets** node in the Project Browser, choosing **Options** to display the Data Set Options dialog box, and setting a **Max Retained** value. When the number of data sets reaches this value, the oldest data set is replaced with any new set saved unless you mark it for non-deletion. You can also rename data sets to organize them.

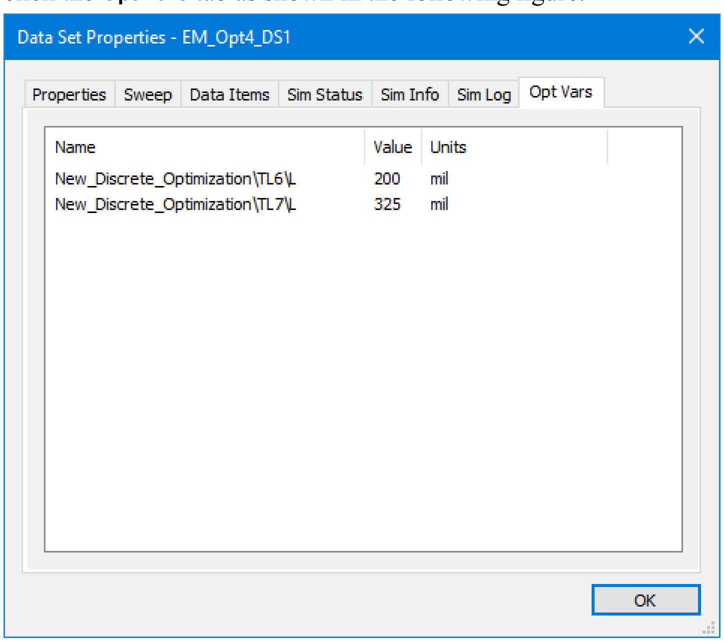
Data sets have slightly different behavior for tuning, optimization and yield analysis.

- Tuning: One new data set is created while tuning.
- Optimization: New data sets are created for each optimization iteration. The data sets display hierarchically as shown in the following figure.



Notice that the data set names start with "Opt". The **Max Retained** value determines how many optimization data sets are kept after optimization is finished. The remaining data sets are only for optimization iterations that resulted in a better optimization cost value. The top data set is the result of the optimization iteration that produced the lowest cost function.

Each data set created during optimization contains the values of the optimization variables used to create that data set. To view these, double-click the data set in the Project Browser to display the Data Set Properties dialog box, then click the **Opt Vars** tab as shown in the following figure.

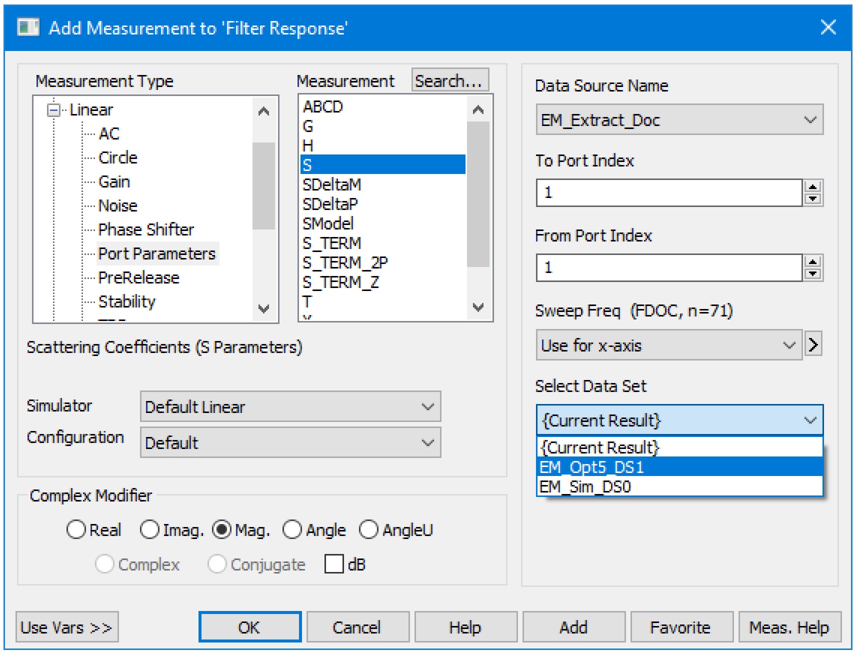


NOTE: The first optimization iteration does not show optimization variables since this simulation is at the initial values of the variables.

- Yield analysis: No new data sets are created for yield analysis. Yield data sets save the data resulting from yield analysis.
- Swept Variables: All of the sweep points are stored in a single data set.

11.3.3. Plotting Directly from Data Sets

After you have data sets for a source document, you can create measurements directly from a data set. When adding a measurement and selecting a **Data Source Name** for a document that has a data set, a **Select Data Set** option displays to allow you to select the appropriate data set. The default is the **{Current Result}** data set, but you can select any available data set.



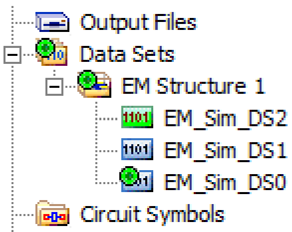
This method works for specific measurements and permanently plots from that data set. There is another mode that allows you to visualize all results for a specific source document from commands directly on the data set. See "Updating Data Sets" for details.

When you plot directly from a data set, to specify that the data set not auto-delete when the **Max Retained** value is reached, you can right-click the data set and choose **Disable Auto Delete**. After you choose this option, the only way to delete the data set is to right-click it and choose **Delete Data Set**. See "Disabling Auto Delete" for details.

11.3.4. Pinning Data Sets

When you choose **Update Results from Data Set**, the next time you simulate, the current data set is used. Sometime you may want to use a different data set when simulating. You can pin a data set that you want to use every time you plot data on a graph. EM data sets are a special case when pinning, when the EM structure is used as a schematic subcircuit or used in extraction. See "Updating and Pinning Specifics" for details specific to EM data sets. When a data set is pinned, edits to the source document do not cause a new simulation to occur since you specified use of a certain set of data.

To pin a data set, right-click the data set in the Project Browser and choose **Pin Results to Document**. All the data set nodes display with a green circle, indicating that there is a pinned data set in use, as shown in the following figure.



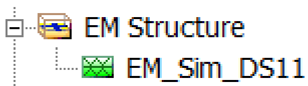
You can only pin one data set at a time. Pinning a different data set moves the pin to that node. You can unpin a data set by right-clicking and choosing **UnPin Results to Document**.

11.3.5. EM Data Set Specifics

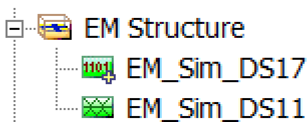
Data sets for EM simulations include the following additional capabilities:

11.3.5.1. Mesh Only Data Set

When you mesh an EM structure, the data is stored in a mesh only data set that displays differently in the Project Browser. If you mesh the structure before simulating it (which is always recommended), you see a mesh only data set as shown in the following figure.



After the EM structure simulates, you see both the mesh and simulation data sets as shown in the following figure.



If you simulate first, you see the simulation data set only. If you do not edit the EM structure and then view the mesh, you do NOT see a mesh data set. The simulation data set in this case also stores the mesh information.

11.3.5.2. Updating and Pinning Specifics

When updated or pinned, EM data sets work slightly differently than other data sets. Any schematic that uses the EM structure as a subcircuit or that creates an EM document using the extraction flow uses the network response from the EM document in its analysis.

NOTE: Any schematic that uses the EM structure as a subcircuit typically does not have a layout that matches what was simulated. Pinning data sets does not update the layout of the EM structure in the project. The green dots on the pinned data set are a visual indication of this information.

By default, when you update from an EM data set, any results from a graph display the data from the chosen data set. Any schematic that uses that EM structure as a subcircuit or for extraction continues to use the current data set.

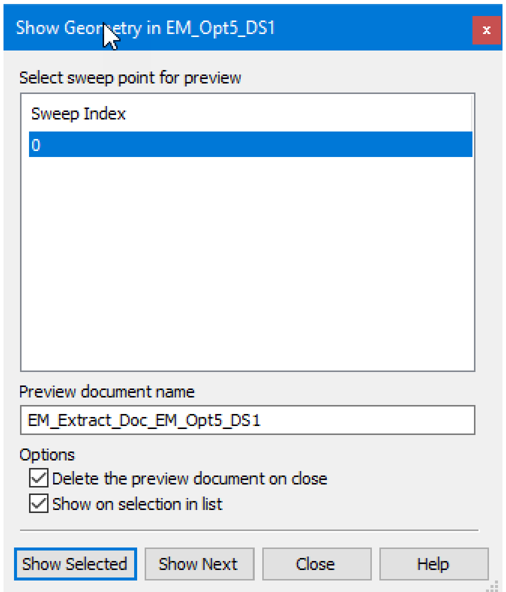
 uoY

can set a mode for the entire project that causes any dependent circuit to resimulate using selected data set results when you update the data set. To toggle this mode, right-click the **Data Sets** node in the Project Browser and choose **Enable Update All EM Dependents** or click the **Include all EM dependents on data set update** button on the Standard toolbar.

Pinning EM data sets cause all schematics that use the EM structure as a subcircuit or extraction to use that data set for the EM results.

11.3.5.3. Viewing Data Set Geometry

Each data set contains all of the information of the structure simulated. You can view the EM structure in the state used to generate the data set by right-clicking the data set and choosing **View Geometry**. The Show Geometry in <data set name> dialog box displays.



This dialog box includes a sweep point section since EM structures can have swept geometry, all of which are stored in one data set. Select the desired sweep and click the **Show Selected** button to generate the EM document used to generate that data set. You can view the stackup parameters and view the 3D layout of the previewed geometry. When you close this dialog box the structure is retained unless you select **Delete the preview document on close**. If this option is cleared, this EM structure is added to your project.

11.3.5.4. Updating Clock if Geometry is Current

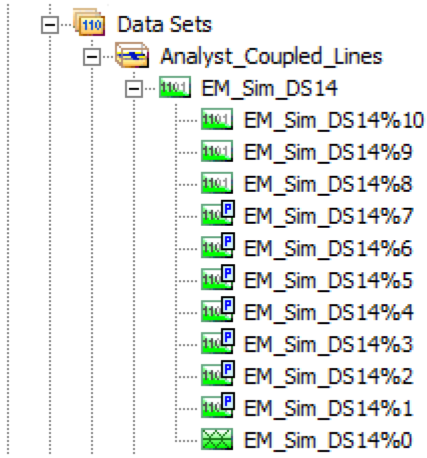
You can right-click a data set that is not clean and choose **Update Clock if Geometry is Current**.

Data set names display in green in the Project Browser when they are current. Each data set contains two clock properties, a Data Clock and a Document Clock. If both clocks have the same values, the data is current and displays in green. Each data set also stores the geometry it simulated to produce the data it is storing. This command compares the current EM geometry with the geometry stored for the data. If they are the same, the two clocks are made equal. Simulation performs the same check and does not simulate if the geometry is unchanged.

Typically, you do not need this command, however some sequences of events can cause the AWR Design Environment software to classify the document as dirty and you can use this command to check. This command is also useful for verifying that data matches the current EM structure geometry when importing data sets into projects.

11.3.5.5. Data Sets for Analyst

Analyst software simulation is slightly different because during the simulation phase, there is data available at each AMR sequence because each sequence produces its own sub-data set. For example, the data sets display similar to the following figure when a simulation runs or is complete.



The sub-data set node icons include a "P" to indicate a Port Only AMR sequence. After the simulation is complete, you can update the data to each sub-data set to see the graph data as well as mesh and any current of field annotations displayed on the 3D view. This is a good way to view the mesh being refined at each AMR step.

NOTE: The sub-data sets are never saved in the project because they can require significant disk space and are typically only needed directly after the simulation is complete, to see its progression. They are still available if you close and reopen the project, however, they are no longer available if you save the project and move it to a new location.

11.3.6. APLAC Data Set Specifics

APLAC simulations generate several different types of data sets, depending on the simulation type. The default name of the data set indicates the type, including:

- DCN_AP_DC: DC simulation
- STB_AP: Stability simulation using the GPROBE2 element.
- LIN_AP: Linear simulation.
- NLN_AP_AC: AC simulation.
- NLN_AP_TR: Transient simulation.
- NLN_AP_HB: Harmonic balance simulation.

When you plot results directly from a data set, you can only choose the type of data set appropriate for the selected measurement.

11.3.7. VSS Data Set Specifics

VSS simulations generate several different types of data sets, depending on the simulation type. The default name of the data set indicates the type, including:

- RFI: RF Inspector simulation
- RFB: RF Budget simulation.
- SYS: Time domain simulation.

When you plot results directly from a data set, you can only choose the type of data set appropriate for the selected measurement.

11.3.7.1. Data Sets for Specific Simulation Type

Only RF Inspector and RF Budget have data sets enabled by default, although projects created before the availability of VSS data sets have data sets turned off for these simulators. To change this behavior you can right-click the **Data Sets** node in the Project Browser and choose **Options** to display the Data Set Options dialog box. Options under **Auto Save Data Sets Options** control data sets for VSS Time Domain, RF Inspector, and RF Budget Analysis.

The VSS Time Domain simulator is not on by default because these simulations run continuously and can therefore produce very large data sets. The **Save VSS TD samples at unused test points** option controls whether data is collected at each node or just nodes where test points are located.

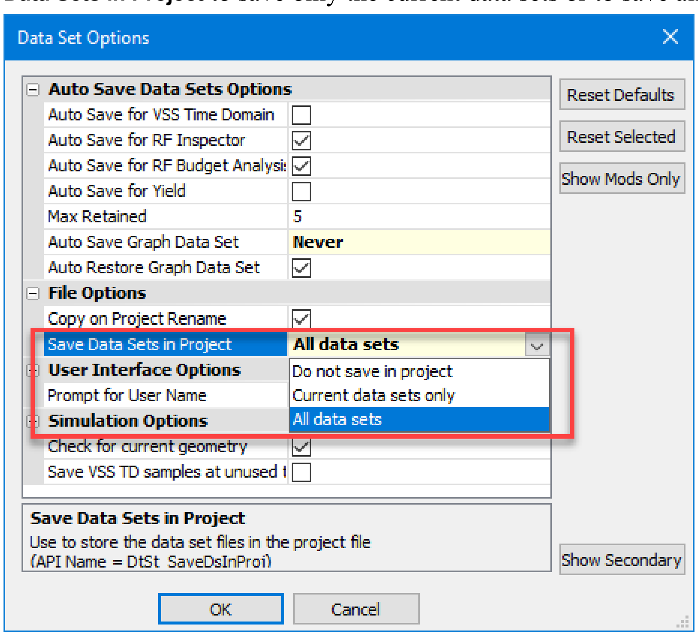
Time domain VSS simulation has two modes: a continuous running mode and a fixed duration mode. Continuous mode is the default when starting, and fixed duration mode usually occurs when specifying that the simulation stop after a certain time, or when setting up swept analysis and setting up bit error rate simulations. In fixed duration mode, when the simulation is done, you have a complete data set and the simulation does not run again until the design is modified. In continuous mode, if you stop the simulation from the controls, the data set is only partially filled. The next time you simulate, a new simulation and data set run whether or not the design is edited.

11.4. Working with Data Sets

The following are common operations for all data set types:

11.4.1. Saving Data Sets in a Project

Data sets are files on disk; they are not saved in a project by default. You can designate that data sets are copied into a project, for example if you want to copy or email the project to a new location. To save a data set with a project, right-click the **Data Sets** node in the Project Browser and choose **Options** to display the Data Set Options dialog box. Change **Save Data Sets in Project** to save only the current data sets or to save all of the data sets.

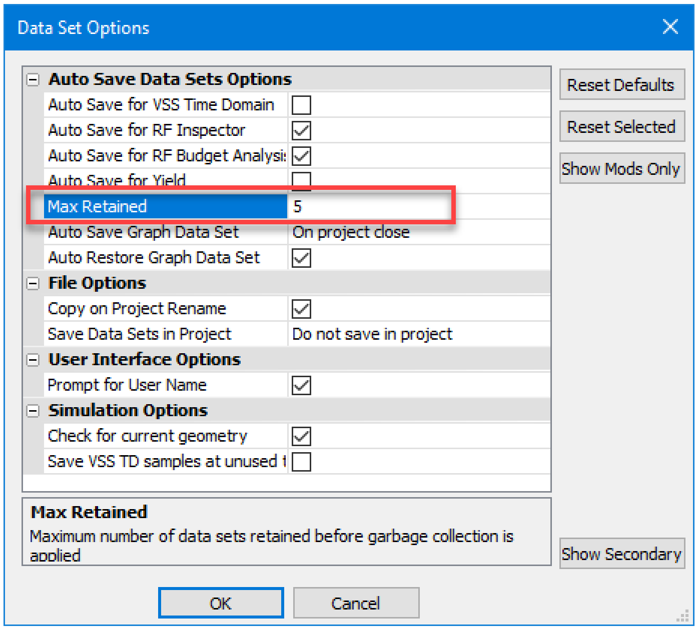


NOTE: Data sets can be very large, especially those from EM simulation if current or fields are stored in the data sets. Saving these in your project can significantly increase the disk space required.

11.4.2. Retaining Data Sets

By default, a project only keeps a specified number of data sets of each type (graph, yield and EM document). When the number of data sets reaches this value, the oldest data set is replaced with any new set saved unless you mark it for non-deletion.

To specify this value, right-click the **Data Sets** node in the Project Browser and choose **Options** to display the Data Set Options dialog box, then specify a value in **Max Retained**.

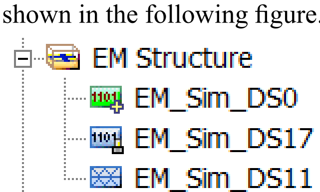


See “Data Set Accumulation” for more information about how the **Max Retained** setting applies to different modes of simulation (tuning, optimizing, yield analysis, and parameter sweeping).

11.4.3. Disabling Auto Delete

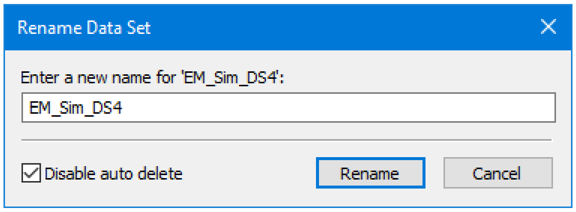
To specify that a specific data set not auto-delete when the **Max Retained** value is reached, you can right-click the data set and choose **Disable Auto Delete**. After you choose this option, the only way to delete the data set is to right-click it and choose **Delete Data Set**. In addition, when renaming a data set, the Rename Data Set dialog box includes a **Disable auto delete** check box you can select.

A data set that is disabled for auto-delete displays a lock in the lower right corner of its icon in the Project Browser, as shown in the following figure.



11.4.4. Renaming Data Sets

You can rename a data set by right-clicking it in the Project Browser and choosing **Rename Data Set**. Type a new data set name in the Rename Data Set dialog box.



Select the **Disable auto delete** check box to disable auto-deletion of the data set when the **Max Retained** value is reached. Typically, renaming a data set indicates an intention to keep the data.

11.4.5. Deleting Data Sets

You can delete data sets just as you would other documents in the Project Browser. You can delete a single data set, you can delete its parent node to delete all for that structure or type, or you can right-click the **Data Sets** node and choose **Delete All Data Sets** to delete all data sets from the project.

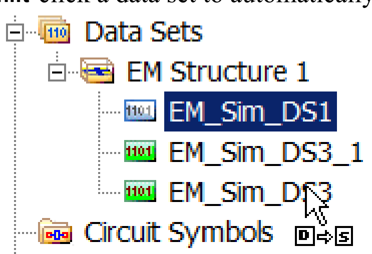
When you delete a source document, all data sets are also deleted, except for any that are marked to disable auto delete.

11.4.6. Updating Data Sets

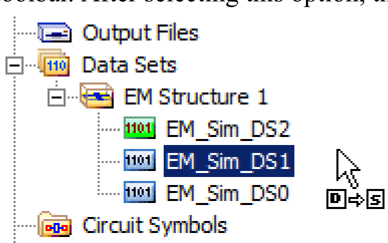
You can update results from a specific data set by right-clicking the data set in the Project Browser and choosing **Update Results** for simulation data sets or **Show Results** for graph and yield data sets. Any graphs that use this data set are then updated.

After you update to a specific simulation data set, you return to the current simulated data by running the simulation again. If nothing changed, the current data sets are used to display the results. If there are any changes, only the necessary simulations occur.

Shift-click a data set to automatically update to those results. The cursor displays as shown in the following figure.



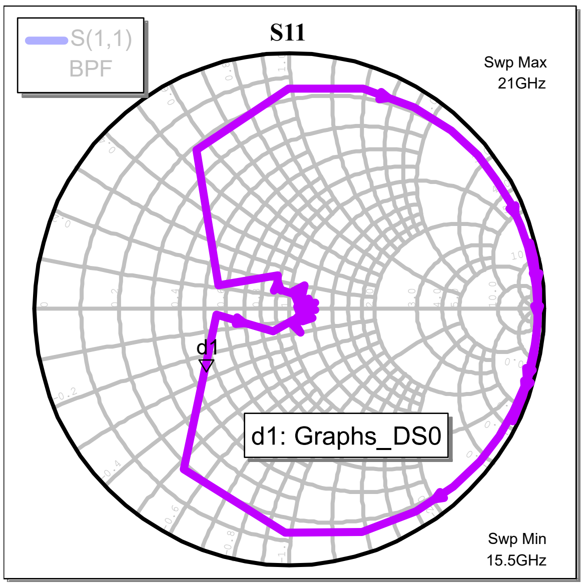
Data sets have a mode that allows you to quickly update the results from many data sets. To enable this mode, right-click the **Data Sets** node and choose **Enable Update Results On Select**, or click the **Data set update on select** button on the Standard toolbar. After selecting this option, the cursor displays differently when hovering near data sets in the Project Browser.



To update results to a specific data set, just select that data set. Select the same commands again to turn off (toggle) this mode.

atD
sets have a mode that allows you to freeze any previous results on a graph when you perform an update. To enable this mode, right-click the **Data Sets** node and choose **Enable Freeze Updates**, or click the **Freeze traces on data set update** button on the Standard toolbar. Select the same commands again to turn off (toggle) this mode. When this command is off, it does not clear any frozen traces from the graph. The graphs no longer accumulate frozen traces from previous data set updates. You can clear the frozen traces from the graph by making the graph the active window and choosing **Graph > Clear Frozen**.

As shown in the following figure, the data on a graph from a data set is different from current simulation results in the following ways:



- A data set marker on the graph shows the name of the data set being plotted. When you simulate to make the data current, these markers no longer display.
- The trace color is different, as specified in the graph trace properties. Each graph has property settings that control how to display different types of traces (frozen, yield, and data sets). You can configure default graph options by choosing **Options > Default Graph Options > <graph type>** and clicking the **Format** tab, or you can set the properties for individual graphs by right-clicking a graph window and choosing **Properties** to display a dialog box with properties for that graph. See "Graph Options Dialog Box: Format Tab" for details.
- The graph legend is grayed out.

11.4.7. Exporting Data Sets

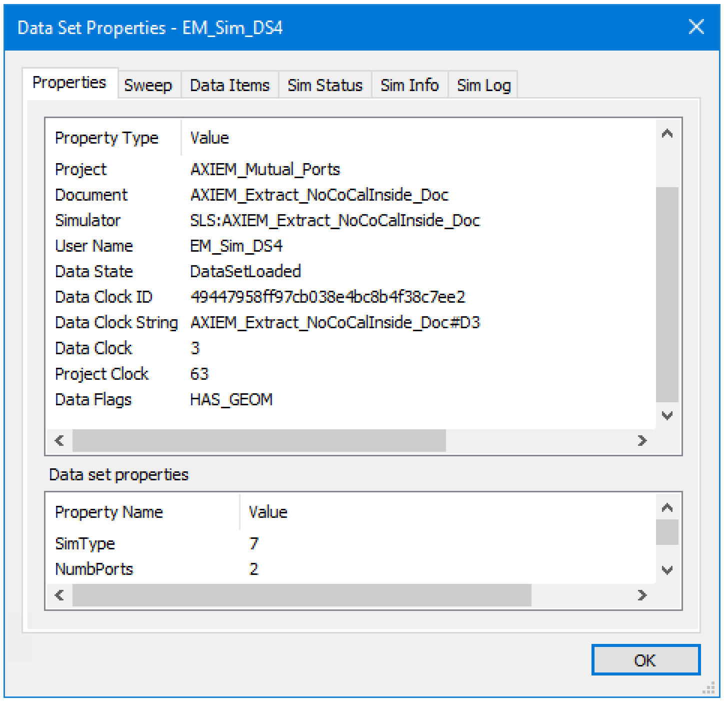
Right-click a data set in the Project Browser and choose **Export Data Set** to display an Export Data Set dialog box that allows you to save the data set to your computer with any file name.

11.4.8. Importing Data Sets

Right-click the **Data Sets** node in the Project Browser and choose **Import Data Set** to display an Import Data Set dialog box that allows you to import any data set on your computer. Graph and yield data sets are easily updated, and new graphs are created to plot the data. See "Using Graph Data Sets in a Blank Project" for details. For simulation data sets, you can create graphs and add measurements to plot the contents of the data sets.

11.4.9. Viewing Data Set Contents

You can double-click a data set in the Project Browser to open the Data Set Properties dialog box and view some of the data set contents.

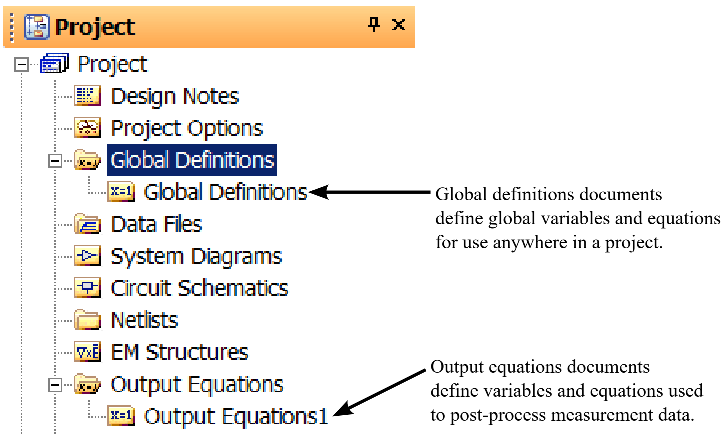


Chapter 12. Variables and Equations

The Cadence® AWR Design Environment® platform allows you to define variables and equations for a number of uses. You can express parameter values within schematics or system diagrams using variables and equations, as well as perform post-processing of measurement data.

12.1. Equations in the Project Browser

The **Global Definitions** node in the Project Browser allows you to define global variables and equations for use anywhere within a project, such as to express parameter values within schematics or system diagrams. In Cadence Microwave Office® software you can also drag any model block (such as a substrate definition) to the Global Definitions window and then reference the material from any schematic. The **Output Equations** node in the Project Browser allows you to define variables and equations used to post-process measurement data. These nodes are shown in the following figure.



12.2. Using Common Equations

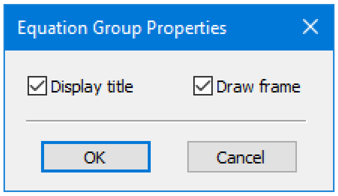
This section includes common equation information for Global Definitions, Output Equations and simulation documents (for example, schematics and system diagrams).

12.2.1. Defining Equations

To define a variable or equation:

1. Double-click to open the desired window and click the window to make it active.
2. Choose **Draw > Add Equation**, click the **Equation** button on the toolbar, or press **Ctrl + E**. A text box displays in the window.
3. Move the text box to the desired location and click to place it there. Begin typing the variable or equation. A list of available variables and built-in functions displays in the equation auto-complete listbox. You can choose any variables or equations in this list by double-clicking or pressing the **Tab** key, or you can continue typing your own. For more information on how to utilize equation/variable auto-complete please see "Equation Auto-Complete". Finish typing the variable or equation, then click outside of the text box or press **Enter** when complete. You can type a multi-line equation by pressing **Ctrl + Enter** to create additional lines. All lines are treated as a group and must contain only one equation per line. Supported variable and equation syntax is provided in "Equation Syntax". You can reference the resulting variable or equation from anywhere in the project.
4. To group and/or title multiple-line equations, right-click on the group and choose **Group Properties** to display the Equation Group Properties dialog box. Select **Display title** and/or **Draw frame** and then click **OK**. Depending on your

selection, a frame displays around the group and "Equation group" displays as a title. Double-click the "Equation group" text to edit the title name.



12.2.2. Editing Equations

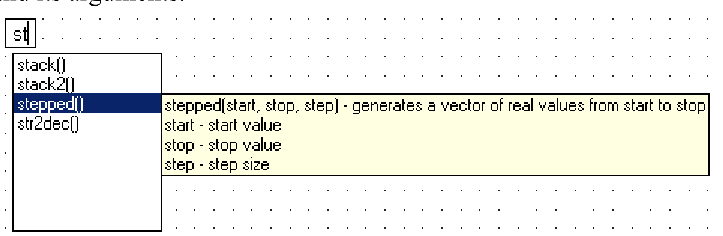
To edit a variable or equation do one of the following:

- Double-click the variable or equation to display a text box. Type the desired changes, then click outside of the text box or press **Enter** when complete. See "Equation Auto-Complete" for more information on how to utilize equation/variable auto-complete.
- Select the variable or equation, then right-click and choose **Properties** to display the Edit Equation dialog box. Alternatively, you can **Shift**+double-click the equation to display the same dialog box. To view this dialog box and a description of its options, see "Edit Equation Dialog Box". Make the desired changes, and click **OK**. For output equations, the Add New Measurement Equation dialog box displays. To view this dialog box and a description of its options, see "Add/Modify Measurement Equation Dialog Box".
- Select the variable or equation, then right-click and choose **Toggle Enable** to alternately disable/enable the equation. Disabled equations are grayed.

12.2.3. Equation Auto-Complete

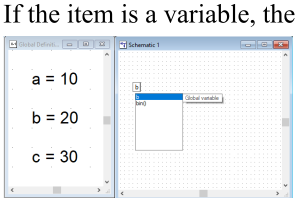
When editing an equation or parameter, a list of available variables and built-in functions displays in a list box. This list is dynamically filtered to show items that match (from the beginning) the word under the edit cursor. You can select items from the list using the up/down arrow keys or by clicking them. Once selected, the item can be used to replace the word under the edit cursor by pressing the **Tab** key or by double-clicking.

As functions are highlighted in the list, a tooltip displays to the right of the item to provide a description of the function and its arguments.



NOTE: Only built-in function display in the list box. User-defined and script-defined functions do not display.

If the item is a variable, the tooltip indicates whether that variable is a local or global variable.



NOTE: Precedence and position of variables is honored. If a local variable is present in the document and above the equation being edited, it is labeled as a local variable. If you are editing above the definition, the label will be global variable. This functions the same as variable evaluations.

12.2.3.1. Filtering

While the auto-complete list box displays you can apply filters to view only functions or variables. To enable filtering, ensure that function-lock is on and then press the following keys:

Hotkey	Filter
F2	Show All/No Filter
F3	Functions Only
F4	Variables Only
F5	Local Variables Only
F6	Global Variables Only

NOTE: Filtering selections resets after closing the edit box for the equation.

12.2.3.2. Turning Off Equation Auto-Complete

You can turn off equation auto-complete by clearing the **Equation edit auto-complete** check box under the **Miscellaneous** group on the Environment Options dialog box. See "Environment Options Dialog Box: Project Tab ".

12.2.4. Displaying Variable Values

You can display the value of an equation by typing the name of the variable followed by the ":" character. When you simulate, choose **Simulate > Update Equations** or press **F6** to update the variable value. The following figure shows a simple equation.

x=2
x:

After an update, the equation displays as follows.

x=2
x: 2

If the displayed variable depends on the measurement being performed (for example, is a function of the frequency sweep) or on the specific instance of a schematic in hierarchy, then the value of the variable is ambiguous, and the displayed value is inconsistent.

12.2.5. Equation Order

Equations use a left-to-right, top-to-bottom sequence. You can use the same variable name multiple times. The final value is the right-most and lowest definition of that variable as shown in the following example.

x=2 x=3
x: 3

x=4
x=5
x: 5

The equation order also matters when creating new equations that use other equations or variables. For example, the following equations are valid and can display a value.

x=2 y=3
x+y: 5

If the order is incorrect, however, equations cannot evaluate and display in red to indicate an error.

x=2
x+y:
y=3

12.2.6. Units for Variables

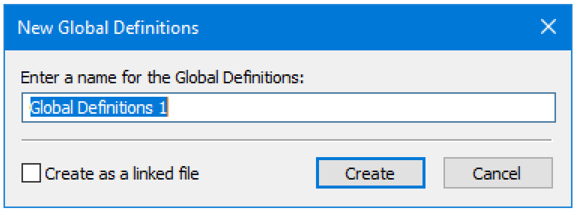
Variable use can be problematic if project units change or if documents are exported and imported into different projects. See “Determining Project Units”" for more information and tips on creating designs that are not sensitive to changing units.

12.3. Using Global Definitions

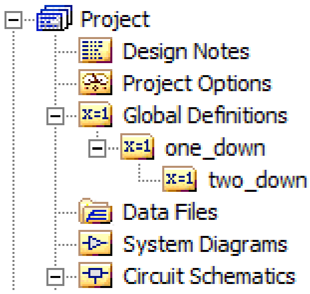
Global Definitions allow you to add and reference multiple pages. Every project includes at least one global definitions page named **Global Definitions**. You cannot rename or delete this document. See “Using Common Equations” for details on adding and editing global variables.

12.3.1. Adding New Global Definitions Documents

To add a new global definitions document, in the Project Browser right-click the **Global Definitions** node or an existing global definitions document and choose **New Global Definitions**. A New Global Definitions dialog box displays to allow you to name the document. To create the document as a linked file , select the **Create as a linked file** check box.



The Project Browser displays the new document one level below the node from which you initiate the command. The following figure shows several levels of global definitions documents.



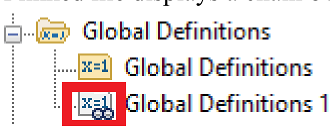
12.3.1.1. Importing Global Definitions Documents

In the Project Browser you can import a global definitions document by right-clicking the **Global Definitions** node or a global definitions document and choosing **Import Global Definitions**. In the Browse For File dialog box, navigate to the global definitions file (*.ieq*) you want to import and click **OK**.

12.3.1.2. Linking to or Embedding Global Definitions Documents

In the Project Browser you can link to or embed a global definitions document in the current project by right-clicking the **Global Definitions** node or a global definitions document and choosing the appropriate command:

- Choose **Link to Global Definitions** to import and link to a global definitions file. Linked documents are not stored in the project file; they are read from file (*.ieq*) when you open the project. The linked file is automatically updated if you change it while the project is open. When you save a project, any unsaved changes are exported to the linked *.ieq* file. A linked file displays a chain overlay on the node icon as shown in the following figure.



- Choose **Link** to convert an embedded global definitions document to a linked document.
- Choose **Embed** to convert a linked global definitions document to an embedded document that is saved in the project file.

12.3.2. Assigning Global Definitions to Simulation Documents

You can assign a specific global definitions document to each simulation document (schematic, system diagram, EM structure, and netlist) and each output equation document. To do so, right-click any of these documents in the Project Browser and choose **Options** to display the Options dialog box. Click the **Equations** tab, and then select a global definitions document from the **Global Definitions** drop-down list.

12.3.3. Global Definitions Search Order

When simulation documents use variables, the following documents are searched for the variables in this order:

1. Simulation document (schematic, system diagram, EM structure, or netlist) using the variable.
2. Global definitions document specified for that simulation document.
3. Next global definitions document up in the hierarchy, until reaching the top of the hierarchy.

An error is issued if the variable is not found.

For example, if a schematic with the same global definitions hierarchy shown previously specified the "two_down" global definitions document and used a variable x, the variable x is first searched for in that schematic, then the "two_down" global definitions document, then the "one_down" global definitions document, and finally the **Global Definitions** document. If the variable is not found after searching in this order, an error is issued.

12.3.4. Renaming Global Definitions Documents

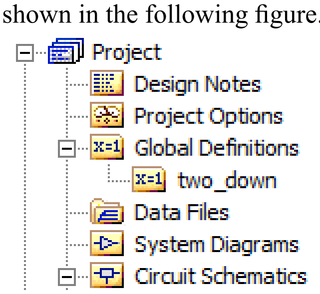
To rename a global definitions document, right-click the document and choose **Rename** to display the Rename Global Definitions dialog box. Select the **Synchronize** check box to update any document that references this global definitions document.

12.3.5. Exporting Global Definitions Documents

To export a global definitions document, right-click the document in the Project Browser and choose **Export Global Definitions**. In the Export Global Definitions dialog box, navigate to the directory in which you want to save the document, and click **OK** to save it as an *.ieq* file.

12.3.6. Deleting Global Definitions Documents

To delete a global definitions document, right-click the document and choose **Delete**, then click **Yes** when prompted to confirm the deletion. You cannot delete the top document. When deleting a node in the middle of a hierarchy, for example the "one_down" document from the previous example, all documents below the current level move up one level, as shown in the following figure.



Simulation documents that point to a deleted global definitions document change and point to the global definitions document one level above the deleted document. For example, a simulation document that points to the now deleted "one_down" document points to the **Global Definitions** level document when "one_down" is deleted.

12.3.7. Defining Global Model Blocks

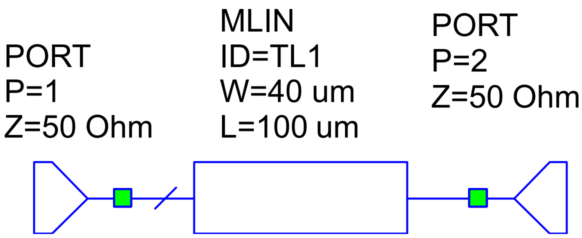
You can add model blocks (such as distributed model substrates, STACKUP blocks, and nonlinear model blocks) to the **Global Definitions** node by selecting and dragging them from the Elements Browser, the same way you add models to schematics or system diagrams. You can also copy from a schematic and paste to the **Global Definitions** node.

12.4. Using Variables and Equations in Schematics and System Diagrams

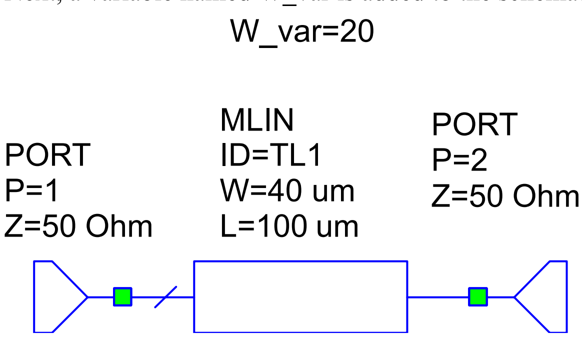
You can define equations and variables locally in schematics and system diagrams. Variables defined in these documents take precedence over the same values defined globally. See "Using Common Equations" for details on adding and editing local variables. The resulting variable or equation is local to the schematic or system diagram and therefore cannot be referenced by any other project component.

12.4.1. Assigning Parameter Values to Variables

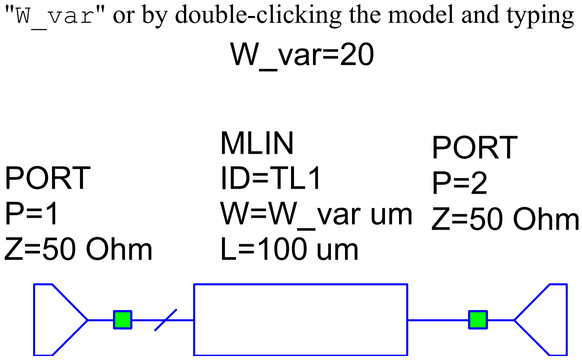
To assign a parameter value to a variable, edit the parameter value as described in "Editing Element Parameter Values" or "Editing System Block Parameter Values", specifying the variable name as its new value. The following example shows a simple MLIN model in a schematic.



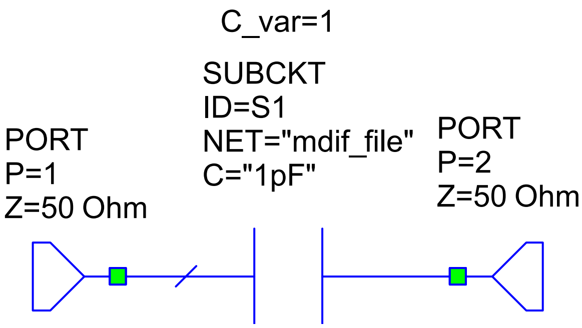
Next, a variable named W_var is added to the schematic.



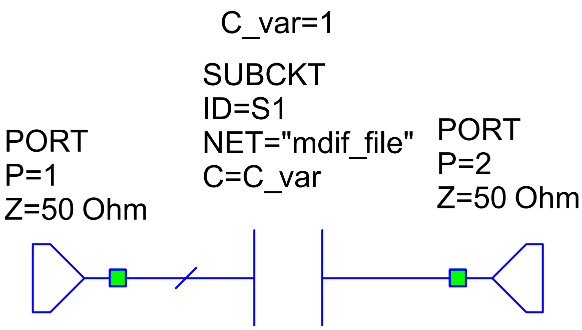
Finally, the parameter W is assigned to this variable by double-clicking the parameter on the schematic and typing "`W_var`" or by double-clicking the model and typing "`W_var`" in the **Value** column for that parameter.



If you use a model that has a list of available parameters, such as an MDIF file, you can assign a variable to a parameter in the same way. For example, see the MDIF subcircuit in the following schematic with the variable C_var defined above it.



You can then assign the parameter to the variable in the same way.



For any model that has parameter lists, the variable value should be a number that is an index into the list of parameters, NOT the actual string value for the model. This is why this variable is C_var=1 instead of C_var="1pF". To remove a variable from this model type, edit the parameter and delete the text completely. This resets the value to first in the available list of values.

12.5. Using Output Equations

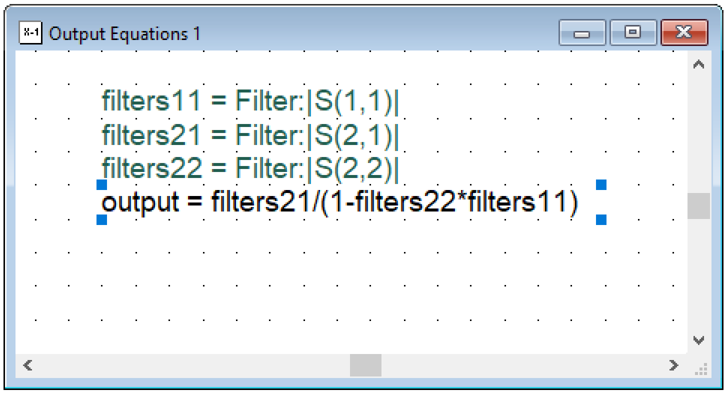
Output equations assign the result of a measurement to a variable, which you can use in other equations just like other variables. A project can include multiple Output Equations documents, each of which can contain multiple output equations and standard equations. Note that the term "output equations" refers to both the type of document, and the type of equations that you can add in those documents.

When editing an output equations document in a window, output equations display in dark green and standard equations display in black. The following example of an output equation defines a variable named "s_data" and assigns to it the magnitude of s11 data from the "Amp1" source document:

```
s_data = Amp1:|S(1,1)|
```

After simulating, the variable contains a vector of real values representing the magnitude of s11 versus its sweep frequency.

In the following Output Equations window, filters11 is a variable that is equal to the measurement s11 of the "Filter" source. Note that the final equation operates on three different measurements.



The units of output equations for various measurements are shown in the following table regardless of the project units you have defined for your project.

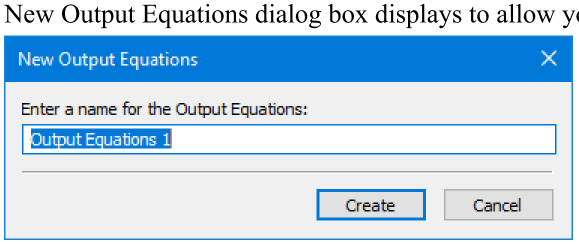
Measurement	Units of Measure
Frequency	Hertz (Hz)
Power	Watts (W)
Voltage	Volts (V)
Current	Amperes (Amp) (Microwave Office)
Phase[a]	Radians (Rad)
Time	Seconds (Sec)
Inductance	Henries (H) (Microwave Office)
Capacitance	Farads (F) (Microwave Office)
Temperature	Kelvin (K)

[a]Radians is an exception to the global unit of measurement set on the Project Options dialog box **Global Units** tab for Angles (degrees).

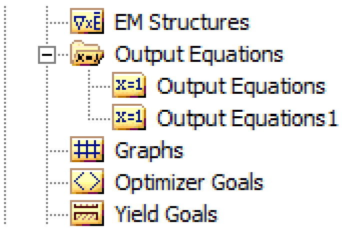
12.5.1. Adding New Output Equations Documents

A new project does not contain any output equation documents, you must add the documents as needed.

To add a new output equations document, right-click the **Output Equations** node and choose **New Output Equations**. The New Output Equations dialog box displays to allow you to name the new document.



The following figure shows the addition of a new output equations document.



12.5.2. Assigning Global Definitions to Output Equation Documents

Since projects can contain multiple global definitions, each output equations document must be assigned a global definitions document to reference if you use global variables. See “Assigning Global Definitions to Simulation Documents” for details on how to assign the proper global definition.

12.5.3. Renaming Output Equations Documents

To rename an output equations document, right-click the document and choose **Rename Output Equations**.

12.5.4. Deleting Output Equations Documents

To delete an output equations document, right-click the document and choose **Delete Output Equations**, then click **Yes** when prompted to confirm the deletion. You cannot delete the default output equations document named **Output Equations**.

12.5.5. Assigning the Result of a Measurement to a Variable

To assign the result of a measurement to a variable:

1. Double-click the output equations document in the Project Browser to display the selected document, then click the window to make it active.
2. Choose **Draw > Add Output Equation**, or click the **Output Equation** button on the toolbar. The Add Measurement Equation dialog box displays. For more information about this dialog box, see “Add/Modify Measurement Equation Dialog Box”.
3. Type a **Variable name**, select a **Measurement Type** and **Measurement** to assign to the variable, specify the required settings, and then click **OK**. The resulting equation sets the variable equal to the measurement data after simulation is performed, and the variable can be used in other standard equations for further processing.

Alternatively, you can easily create an output equation from a measurement on an existing graph:

1. Double-click the output equations document in the Project Browser to display the selected document, then click the window to make it active.
2. In the Project Browser, select the desired measurement and drag it to the open Output Equations window, then release the mouse button and click to place the new output equation. The equation is given a default name such as "EQ1". When you add output equations using this procedure ensure that the name assigned to the variable is unique.

12.5.6. Editing Output Equations

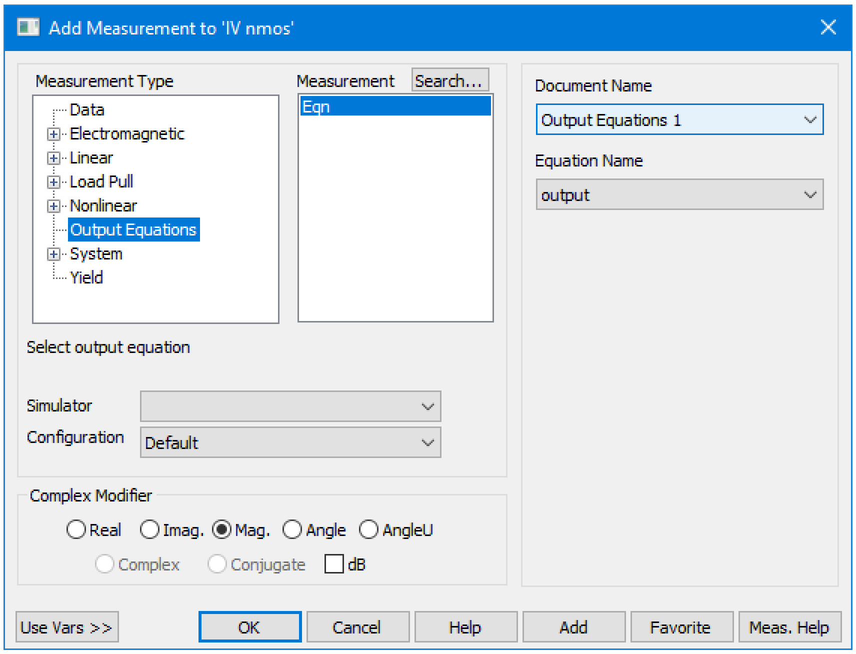
To edit an output equation, right-click the equation and choose **Properties**. Cadence does not recommend trying to edit the text string directly by double-clicking on these equations. The dialog box to choose the specific measurement ensures the measurement syntax is correct.

To edit standard equations in an output equations document, follow the instructions in "Editing Equations". See "Equation Syntax" for supported variable and equation syntax.

12.5.7. Plotting Output Equations

You can plot any equation in an output equations document like regular measurements. To plot such an equation:

1. Right-click an existing graph and choose **Add Measurement**. A dialog box similar to the following displays.



2. Select **Output Equations** as the **Measurement Type**. Select **Eqn** as the **Measurement**.
3. Select the appropriate output equation document from the **Document Name** field.
4. In **Equation Name**, select the equation or variable you want to plot.
5. Equations can result in complex values, so choose the desired complex modifier settings and then click **OK**.
6. When you simulate, the results display on the selected graph.

NOTE: When plotting equations representing impedance or admittance data on a Smith Chart, you must normalize the data yourself. For example, if you have impedance data, you transform by (Z-Z0)/(Z+Z0) first. This differs from plotting a measurement directly on a Smith Chart, where the system knows what type of data the measurement represents and automatically performs the transformation. Automatic transformation only occurs for Smith Charts.

12.6. Using Scripted Equation Functions

Scripted functions allow you to extend the functions that you can reference in equations, by referencing functions written in BASIC script. In earlier AWR Design Environment platform versions, equations were limited to functions that were intrinsic to the application and there was no convenient way to add or expand functions that were unique to the project.

The following sections for BASIC users describe how to use BASIC scripts to add customized functions to an AWR Design Environment platform project.

12.6.1. Adding Equation Functions

Scripted functions called from an equation must follow the same scripting guidelines as any other equation. The only difference with equation functions is that they must exist in a code module named "Equations".

When the AWR Design Environment software loads a project, it looks for the Equations module and identifies the functions so they can be referenced from an equation. Functions defined in other modules cannot be referenced from an equation.

You can define a function to take any number of parameters of varying types including strings, integers, doubles, complex and variants. You can define data arrays of variable length, or no parameters at all. The following is an example of passing parameters to a function.

```
' This function take a string and a double
Function Example(X As String, Y as Double)

' This function takes an array of doubles and a
' complex number

Function Example(X() As Double, Y As Complex)

' This function takes an array of complex numbers
and
' a Variant
Function Example(X() As Complex, Y As Variant)

' This function does not take any parameters
Function Example()
```

A function always returns a single value which can be defined as a specific type. If the return type is omitted, the function returns a variant. The following is an example of functions defined to return a specific type.

```
' This function returns a string
Function Example() As String

' This function returns a double
Function Example() As Double

' This function returns a variant
Function Example() As Variant

' This function returns a variant by default
Function Example()

' This function returns a complex number
Function Example() As Complex

' This function returns an array of doubles
Function Example() As Double()
```

Functions can also return arrays. An array can be useful if your function needs to return multiple values. The following is a function that returns an array of doubles.

```
Function Example() As Double()

 ' Set each value to an element in the array
 Dim Result(3) As Double
```

```
 Result(0) = 1.2
 Result(1) = 3.4
 Result(2) = 5.6
 Example = Result

End Function
```

12.6.2. Referencing a Function in an Equation

You reference scripted functions in equations in the same way you reference intrinsic functions. Use the name of the scripted function and pass the parameters as follows:

```
' Scripted function for calculating the circumference
of a circle

Function Circumference(r As Double) As Double
  Circumference = 3.14159 * 2 * r

End Function

' AWRDE Equation
referencing the
Circumference function

X = Circumference(1.6)

X: 10.05
```

The following table shows various function definitions and the equation syntax used for passing the parameters.

Passing a string	`Function Func1(Value As String)`
	`X = Func1("hello world")`
Passing an array of strings	`Function Func2(Value() As String)`
	`X = Func2({"a", "b", "c"})`
Passing a double	`Function Func3(Value As Double)`
	`X = Func3(1.2345)`
Passing an array of doubles	`Function Func4(Value() As Double)`
	`X = Func4({1.23, 4.56, 7.89})`
Passing a string and a double	`Function Func5(Value1 As String, Value1 As Double)`
	`X = Func5("The area is ", 9.9)`
Passing a complex number	`Function Func6(Value As Complex)`

	`X = Func6(2.33+j*3.45)`
Passing an array of complex numbers	`Function Func7(Value() As Complex)`
	`X = Func7({1.23+j*12, 4.56+j*34, 7.89+j*56})`

12.6.3. Local and Global Scoping

The AWR Design Environment platform provides two levels of scoping for functions: global and local. You can use a globally scoped function in all of your AWR Design Environment platform projects. A function that is scoped locally, however, can be used only in the project in which it was created. Therefore, to define a function for use in several projects, you should scope it globally; to define a function that is specific to the current project, you should scope it locally.

To scope a function globally, add it the **Equations** code module of the global script located in the Scripting Editor (choose **Tools > Scripting Editor** under **Global > Code Modules**). If the **Equations** module does not exist, you can create it by right-clicking **Code Modules** and choosing **Insert Module**. Right-click the new module and choose **Rename Module1** to name the module "**Equations**". To scope a function locally, add the function to the **Equations** module for the open project.

12.6.3.1. Local Versus Global Functions

If two functions have the same name, for example one in the global module and one in the local module, the function in the local module takes precedence and is executed when referenced from an equation. In this case, a project overrides globally defined behaviors to meet the project needs.

For example, to define a function named 'Correct' for use in all of your projects (it multiplies an input parameter by a correction factor), add a function to the **Global Definitions** module as follows, and reference it in several equations of your projects.

```
' Global Correct function

Const cf = 1.12

Function Correct(Value As Double) As Double

 Value = Value * cf

End Sub
```

If one of the projects using this function needs a different correction factor, you could create a new local function called 'Correct1' with the new correction factor, and then replace 'Correct' with 'Correct1' in all of the equations that reference the 'Correct' function, however this may not be a good solution if there are many references to the original 'Correct' function, and because you might miss a reference when changing the name.

A better solution is to add a local function using the same name as the global function. Because the local function takes precedence over the global function, the equations referencing the 'correct' function use the new local function and do not need modification.

```
' Local Correct function which overrides the Global
function

Const cf = 0.98
```

```
Function Correct(Value As Double) As Double

       Value = Value * cf

End Sub
```

12.6.4. Scripting and Debugging Tips

Scripted equation functions use standard BASIC and standard scripting techniques.

12.6.4.1. Scripting Functions to Call Other Functions

If you have a routine or calculation common to several functions, you can break it down into a separate subroutine that you can call from each function that uses its functions. The following example shows two Adjust functions that call a common DoCalc routine:

```
Function AdjustDouble(Value As Double) As Double
  Value = Value * 2
  Adjust1 = DoCalculation(Value)

End Function

Function AdjustHalf(Value As Double) As Double
  Value = Value / 2
  Adjust1 = DoCalculation(Value)

End Function

Function DoCalculation(Value As Double) As Double
  If Value > 1 Then
   DoCalculation = Value ...
  Else
   DoCalculation = Value ...
  End if

End Function
```

12.6.4.2. Using 'Debug.Print' To Verify Results

You can analyze the 'Debug.Print' statement to verify that the function is performing as expected. You can check input values, intermediate results, and output values.

To view the output of the 'Debug.Print' statement:

1. Open the Scripting Editor and choose **View > Split Window**.
2. Click the **Immediate** tab.

The following example shows code using the Debug.Print statement to display the values of an input array, the intermediate results of the total, and the final value the function returns.

```
Function SumValues(Values() As Double) As Double

 Dim Total As Double

 Total = 0

 ' Loop through the array and total the values
```

```
 For Index = LBound(Values) To UBound(Values)

  ' Debug the input parameters

  Debug.Print "Double(" & Index & "): "_

      & Values(Index)

  ' Debug the Total as it is calculated

  Total = Total + Values(Index)

 Next Index

 ' Set the return value

 SumValues = Total

 'Check the value that will be returned

 Debug.Print "SumValues is returning: " & Total

End Function
```

12.6.4.3. Setting Breakpoints to Inspect Variables

For complex functions, the 'Debug.Print' statement may indicate a problem, but not the cause. In these cases you may need to review each line. You can do so by setting a breakpoint in the function so that the next time the referencing 'Equation' is calculated, execution of the function stops on the line of code where the breakpoint is set. With the function execution paused, you can look at the values of all the function variables and step through the rest of the function line by line to identify the problem. Click **Run** when you want to resume normal execution.

12.6.4.4. Creating a Test function to Validate Results

You may find it useful to create a special function to validate the results of your equation functions by passing known values. It is good practice to test your functions by passing values at the outer edges of the expected range as well as values in the middle of the range and known transition points. You can call this special function from `Sub Main` to allow the test to be performed independently of equation calculations, as shown in the following example.

```
' Code Module

Sub Main
  TestFunc1
  TestFunc2 ' Not shown
  TestFunc3 ' Not shown

End Sub

' Test Func2 with known values
Sub TestFunc1
  ' Using debug output

  Debug.Print Func1(-1.0)
  Debug.Print Func1(0.0)
  Debug.Print Func1(5.0)
  Debug.Print Func1(10.0)
  Debug.Print Func1(11.0)
```

```
  'By displaying a message box
  If Func1(6) <> 43 Then
    MsgBox "Func1 did not return the correct value"

  End If

End Sub

' Expected range for Value is 0-10
Function Func1(Value As Double) As Double
  If Value < 0 Or Value > 10 Then
   Func1 = -1.0
  Else
   Func1 = Value * Value + 7.0
   End If

End Function
```

12.7. Equation Syntax

The basic form for an equation consists of the variable name on the left side of the assignment operator and a mathematical expression on the right. The syntax of the expression follows the general rules of algebra.

If the expression is not valid, the equation displays in green and an error displays in the Error window. If the equation is not visible, double-click the error in the Error window to display it.

12.7.1. Operators

You can use the following operators:

Operator	Precedence Level	Description
+	1	Unary positive
-	1	Unary negative
!	1	Logical not
~	1	Bitwise not
^	2	Power of
*	3	Multiply
/	3	Divide
%,mod	3	Modulus
+	4	Add
-	4	Subtract
sll	5	Shift Left Logical
srl	5	Shift Right Logical
<	6	Less than
>	6	Greater than
<=	6	Less than or equal to
>=	6	Greater than or equal to

Operator	Precedence Level	Description
==	6	Equality
!=	7	Inequality
&,and	8	bit-wise AND
nand	8	bit-wise NAND
xor	9	bit-wise exclusive OR
xnor	9	bit-wise exclusive NOR
|,or	10	bit-wise OR
nor	10	bit-wise NOR
&&	11	logical AND
||	12	logical OR
=	13	Assignment
:	13	Display value
<<	13	Declare external parameter

12.7.2. Variable Definitions

You can make assignments to constants or to mathematical expressions:

```
A = 6
val = x * (4 + y)
```

Variable names must follow these rules:

- The variable name may contain the characters A-Z, a-z, 0-9, and "_" (underscore), with no spaces.
- The first character of the variable name may not be a digit.
- The variable name is case-sensitive.

12.7.2.1. Function Definitions

You can define functions by following the variable name with a list of arguments enclosed in parentheses. The following example creates a function that takes two arguments:

```
sum( a, b ) = a + b
```

You use this function by replacing the arguments "`a`" and "`b`" with any valid mathematical expression. The following equation assigns variable "`c`" to (-3) + (2*4) = 5:

```
c = sum( -3, 2*4 )
```

12.7.2.2. Representing Complex Numbers

You can assign a variable to a complex number by using the imaginary global constant "i" or "j" as a multiplier:

```
complexZ = 50 - j*1.3
```

12.7.2.3. Array Indexing

An array is an n-dimensional data set, where n is 0-, 1-, or 2-dimensional. The data type must be real, complex, or string.

- Scalar - 0-dimensional array
- Vector - 1-dimensional array
- List - 1-dimensional array
- Matrix - 2-dimensional array

Array indexes are 1-based. If a floating-point value is used as an array index, it is first truncated to an integer.

Arrays may be sliced by using an index wildcard '*':

`x` - returns whole array

`x[*]` - returns whole 1D array

`x[*,*]` - returns whole 2D array

`x[*, 1]` - returns first column in 2D array

`x[3, *]` - returns third row in 2D array

Indices can also be vectors, and therefore use any vector syntax.

Array Indexing Examples:

```
x[{1, 3, 5}]
```

returns the first, third, and fifth elements of an array.

```
x[stepped(1, 10, 1)]
```

returns the first 10 elements

```
x[stepped(1, vlen(x), 2)]
```

returns every other element in the array

12.7.2.4. Precedence

Common algebraic precedence applies to mathematical expressions. The order in which equations are arranged on the screen determines their computation precedence. For a variable to be used in another equation it must come earlier in the computation order as follows:

- Equations that are lower on the page are placed after ones that are higher.
- If two equations are on the same line on the page, the one to the right is placed after the one to the left.

If a variable name referenced in an equation is defined more than once, the value taken is from the one most recently defined before the referencing equation. In the following example "`b`" has a value of 2 and "`c`" has a value of 3.

```
a = 1  a = 2  b = a
a = 3
c = a
```

12.7.3. Built-in Functions

You can use built-in functions from the following categories.

MATH OPERATIONS

The following functions can be performed on a matrix, vector, or scalar. They can be performed on real and complex numbers unless otherwise specified.

Function	Description
`arccos(x)` or `acos(x)`	inverse cosine of `x`, with result in radians
`arcsin(x)` or `asin(x)`	inverse sine of `x`, with result in radians
`arctan(x)` or `atan(x)`	inverse tangent of `x`, with result in radians
`atan2(y, x)`	four quadrant inverse tangent of `y/x`, with result in radians. `x` and `y` must be real.
`acosh(x)`	inverse hyperbolic cosine of `x`, with result in radians
`asinh(x)`	inverse hyperbolic sine of `x`, with result in radians
`atanh(x)`	inverse hyperbolic tangent of `x`, with result in radians
`cos(x)`	cosine of `x`, with `x` in radians
`sin(x)`	sine of `x`, with `x` in radians
`tan(x)`	tangent of `x`, with `x` in radians
`cosh(x)`	hyperbolic cosine of `x`, with `x` in radians
`sinh(x)`	hyperbolic sine of `x`, with `x` in radians
`tanh(x)`	hyperbolic tangent of `x`, with `x` in radians
`ceil(x)`	Returns the smallest integer value not less than `x`. `x` must be real.
`erf(x)`	error function at `x`
`erfc(x)`	complementary error function at `x`
`exp(x)`	natural exponent of `x`
`expm1(x)`	e raised to the power of `x-1`
`floor(x)`	Returns the largest integer value not greater than `x`. `x` must be real.
`fmod(x, y)`	Returns the floating point remainder of `x` divided by `y`. `x` and `y` must be real.
`heaviside(x)`	Returns 1 if `arg` is greater than or equal to 0, 0 otherwise. `x` must be real.
`hypot(x, y)`	Euclidean distance between `x` and `y`. `x` and `y` must be real.
`int(x)`	Returns the truncated integer value of `x`. `x` must be real.
`integrate(x)`	Calculates the area under the curve for the data in `x`. `x` must be real, contain at least two points, and have swept data. Uses trapezoidal rule.
`lgamma(x)`	Calculates the natural logarithm of the absolute value of the gamma function at `x`.
`log(x)`	natural logarithm of `x`

Function	Description
`log10(x)`	base 10 logarithm of `x`
`log1p(x)`	natural logarithm of `x+1`
`pow(x, y)`	`x` raised to the power of `y`
`round(x, s)`	Returns the rounded number of `x` with step size `s`. `x` must be real.
`sign(x)`	Returns 0 if `x` is equal to 0, 1 if `arg` is greater than 0, -1 otherwise. `x` must be real.
`sinc(x)`	Returns sin(x)/x for non-zero x, returns 1 if x is zero.
`sqrt(x)`	square root of `x`
`tgamma(x)`	Calculates the value of the gamma function at `x`.
`trunc(x)`	Rounds `x` to the nearest integer towards zero.

TYPE CONVERSION

The following functions can be performed on a matrix, vector, or scalar. Any argument of type "str" is a string and any other arguments must be real values.

Function	Description
`bin(str)`	Returns the decimal representation of the binary value represented by `str`.
`cint(x)`	Returns `x` as a rounded integer. `x` must be real, and can be scalar or array.
`complex(real, imag)`	Returns a complex number made from `real` and `imag`.
`cstr(x)`	Returns `arg` as a string. `x` must be real, and can be scalar or array.
`dbpolar(dbMag, ang)`	Returns a complex number made from 10^(dbMag/20) and `ang` in degrees.
`dec2binvect(dec,n)`	Converts a `dec` (decimal) value to a binary number. Returns an integer vector with minimum length of `n` bits.
`hex(str)`	Returns the decimal representation of the hexadecimal value represented by `str`.
`oct(str)`	Returns the decimal representation of the octal value represented by `str`.
`polar(mag, ang)`	Returns a complex number made from `mag` and `ang` in degrees.
`str2dec(str, base)`	Converts the string `str` into a numeric value using `base` as the representation base. For example, if `str` is 123 and `base` is 10 (decimal), the generated value is 123. If `base` is 8 (octal) the value is 83. If `base` is 16 (hexadecimal) the value is 291.

UNIT CONVERSION

The following functions can be performed on a matrix, vector, or scalar. They can be performed on real numbers only.

Function	Description
`awg_dia(x)`	Returns wire diameter in meters given gauge. Valid gauge values are: 0, 1, 2, 4, 6, 8, 10, 12, 13, 14, 16, 18, 20, 22, 24, 25, 26, 28, 30, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 44, 46
`ctof(x)`	Returns `x*9/5+32.0` where `x` is in degree Celsius
`ctok(x)`	Returns `x+273.15` where `x` is in degree Celsius
`db(x)`	Returns `20*log10(x)`

Function	Description
`db_pow(x)`	Returns `10*log10(x)`
`deg(x)`	Returns `x` multiplied by `180/pi` (converts radians to degrees)
`ftoc(x)`	Returns `(x-32)*5/9` where `x` is in degree Fahrenheit
`ftok(x)`	Returns `(x-32)*5/9+273.15` where `x` is in degree Fahrenheit
`ktoc(x)`	Returns `x-273.15` where `x` is in degree Kelvin
`ktof(x)`	Returns `(x-273.15)*9/5+32.0` where `x` is in degree Kelvin
`lin(x)`	Returns 10^(x/20)
`lin_pow(x)`	Returns 10^(x/10)
`rad(x)`	Returns `x` multiplied by pi/180 (converts degrees to radians)

VECTOR OPERATIONS

The following functions can be performed on a matrix, vector, or scalar. They can be performed on real and complex numbers unless otherwise specified.

Function	Description
`amax(x)`	Returns the maximum value of array `x` over all dimensions (all traces). `x` must be real.
`amax1(x)`	Returns the maximum value of array `x` over the first dimension (per trace). `x` must be real.
`amin(x)`	Returns the minimum value of array `x` over all dimensions (all traces). `x` must be real.
`amin1(x)`	Returns the minimum value of array `x` over the first dimension (per trace). `x` must be real.
`asize(x)`	Returns the size of an array for an output equation measurement. This function returns two values: the first is the number of points on the x-axis and the second is the number of swept traces for the measurement `x`.
`assign_array(array, index, value)`	Assigns `value` to the entry in the `array` at `index`. Note that `index` and `value` can be arrays thus allowing assignment of multiple values at once. The type of `array` and `value` must be the same. The size of `index` and `value` must be the same.
`asum(x)`	Returns the sum of all elements in array `x`. `x` must be real or complex.
`concat(a, b)`	Concatenates vectors `a` and `b`. Arguments must have the same type.
`csum(x)`	Returns the cumulative sum of `x` versus the swept data of `x`
`deltawin(y, swpval1, swpval2, winwidth, winpos)`	Returns a vector of the difference between the maximum and minimum y values across a moving window. The difference is calculated as the maximum to minimum change in y data over the x value sweep range specified by winwidth. swpval1 and swpval2 define the minimum and maximum swept x value range and must be positive values. winwidth must be ≤ (swpval2 - swpval1). For winpos=1 (extend right) the window ranges from x[i] to x[i]+winwidth, for winpos=2 (center) the window ranges from x[i]-winwidth/2 to x[i]+winwidth/2, and for winpos=3 (extend left) the window ranges from x[i] to x[i]-winwidth.
`der(x, y)`	Returns the finite difference derivative of `x` with respect to `y`. Arguments must be real scalars or real vectors. If both arguments are vectors, then they must be of equal size.
`fill(n, val)`	Returns a 1-dimensional array of length `n` with all values equal to `val`.
`find_index(x, val)`	Returns the index of a value in array `x` which is closest to `val.`

Function	Description
find_index_range(x, val1, val2)	Returns a list of indices {i1,...,i2} where i1 corresponds to the index whose value in array x is closest to val1, and i2 corresponds to the index whose value in array x is closest to val2.
histogram(x, bin_type)	Computes the histogram of data values represented in x. x must be a real vector. The bin_type parameter controls how the data is binned. bin_type<0 means bin_type specifies the number of bins. bin_type=0 means the number of bins is auto-calculated using number_of_points/ln(2.0). bin_type>0 means bin_type specifies the width of a bin.
interp(type, x, y, new_x)	Returns interpolated y data at new X points. x, y, and new_x must be real vectors, and x and y must be the same size. type can be 0 (linear), 1 (polynomial), 2 (rational), or 3 (cubic spline).
interp_poly(order, x, y, new_x)	Returns interpolated y data at new X points using polynomial interpolation. x, y, and new_x must be real vectors, and x and y must be the same size. order is the order of the polynomial and can be between 1 and 100 inclusive.
lin_reg(x)	Returns a least-squares-fit line to x. x must be a real vector. The returned data is versus the original swept data of x.
max(a, b)	Returns a vector of value a[i]>b[i]. Arguments must be real scalars or real vectors. If both arguments are vectors, then they must be of equal size.
min(a, b)	Returns a vector of value a[i]<b[i]. Arguments must be real scalars or real vectors. If both arguments are vectors, then they must be of equal size.
mmult(x, y)	Returns the matrix product of matrices x and y.
mov_avg(x, n)	Calculates the moving average of x using a window size of 2*n+1. n is the number of points on either side of the current point. x may be real or complex and n must be a real scalar that is rounded to an integer. This can be used to smooth a graph. For complex x, averaging is performed on real and imaginary parts (for example, on Cartesian coordinates, rather than polar).
stack(n, vec)	Returns a 2-dimensional array whose size is n by vlen(vec). The values of the array are the values of vec stacked up n times.
stack2(n, vec)	Returns a 2-dimensional array ("matrix") whose size is n by vlen(vec)/n. vec is split into n equal size pieces to create the matrix columns.
transpose(x)	Transposes x which contains 2D data.
unwrap(x, d)	Unwraps phase. x represents the swept data (usually phase) and must be real. d is the threshold that causes an offset of 2*d to be added to the data if there is more than d change from one point to the next. d must be real and scalar. This function will maintain the sweep data for x if it exists.
vlen(x)	Returns the length of the vector x.

ADDITIONAL FUNCTIONS

Function	Description
assign_sweep(x, y)	Assign sweep data of y to x.
assign_swpunit(x, unitType)	Returns x with a unit type assigned to its sweep data. unitType must be a real scalar in the range of 0 to 16 with values of: 0 (none), 1 (frequency), 2 (capacitance), 3 (inductance), 4 (resistance), 5 (conductance), 6 (length: metric), 7 (length: English), 8

Function	Description
	(temperature), `9` (angle), `10` (time), `11` (voltage), `12` (current), `13` (power: log), `14` (power: linear), `15` (dB), `16` (string).
`circle(radius, ctr_re, ctr_im, num_pts)`	Returns complex data representing a circle of given `radius` and center point (`ctr_re`, `ctr_im`) with `num_pts` number of points. `num_pts` must be between 2 and 1000 inclusive.
`col()`	See row() and col().
`DataFile(name) or DataFile(name, "c")`	Returns an array of the Data File `name` from the Project Browser. `DataFile(name)` returns a real array and `DataFile(name, "c")` returns a complex array. *You cannot place these functions in Output Equation documents.* If the imported file data is needed there, you can write the DataFile(..) function in a Global Definitions document.
`DataFileCol(name, columnName)`	Returns the column with name `columnName` of the Data File `name` from the Project Browser. If the column contains real data, a vector of real values is returned. If the column contains complex data, a vector of complex values is returned.
`data_file(name, type, args)`	Converts the values in the data file `name` into a numeric vector. `Type` and `arg` parameters are reserved for future use and should be set to "". See "vfile".
`find_pv(val, e_series)`	Finds the nearest preferred value to `val` as defined by the IEC60063 standard. `e_series` may be 3, 6, 12, 24, 48, 96, or 192. Non-positive values are returned unchanged. This function is typically used to find standard values for resistors, capacitors, and inductors.
`if(cond, trueval, falseval)`	If `cond` is true, then evaluates to `trueval`, otherwise evaluates to `falseval`. `cond` must be real and scalar. cond may use equation operators such as Y = if(X==0,1,2). (See "Operators" for information about equations operators.) If vectors, the lengths of `cond`, `trueval`, and `falseval` must match.
`marker(graph, mN)`	Returns a vector of real values corresponding to the values displayed on marker `mN` in `graph`. The arguments are text and must be enclosed in quotation marks. For example, `marker("Gain","m2")` returns values on marker "m2" in graph "Gain". If units are known, the values are in base units; for example, frequency is in Hz. Complex values are returned as the two real values displayed on `mN`, which depend on properties set on `graph`. This function updates automatically if `mN` is moved.
`plot_vs(y, x)`	Plots the array `y` versus the vector `x`. `x` must be real and the first dimension of `y` must match the length of `x`.
`plot_vs2(y, x, unitType)`	Plots the array `y` versus the vector `x` with the `unitType` set. `x` must be real and the first dimension of `y` must match the length of `x`. See "assign_swpunit" for unit types.
`row(), col(), row(data_var_name, row#), col(data_var_name, col#)`	Accesses any row or column of data in a text data file. The data file must be in text format and must not contain any header information. Data can be column, space, or white space delimited. Before using `row()` and `col()` functions, the target data file must be "declared" using the `DataFile()` function: `Data_Var_Name = DataFile("file_name")`. After the data file is declared, the `Row()` and `Col()` functions access the specified row or column in the data file: `Row_Var_Name = Row(Data_Var_Name, Row#) Col_Var_Name = Col(Data_Var_Name, Col#)` **NOTE:** The `DataFile()` function uses a text string to identify its argument, so the file name is in quotes. Once declared, the `Data_Var_Name` is just a variable, so it is not in quotes when used in the `Row()` and `Col()` functions.
`subsweep(meas, x1, x2)`	Returns the measurement `meas` over the sweep range.

Function	Description
`subsweepi(meas, start, count)`	Returns the measurement `meas` from `start` to `start + count`, where `start` and `count` are integer numbers referring to the vector index.
`swpunit(x)`	Returns the unit type (see "assign_swpunit") for the sweep data of `x`.
`swpvals(x)`	Returns the sweep data of `x`. `x` is real.
`vfile(name)`	Converts values in the text data file `name` into a numeric vector. If the text file contains one value per line, vfile `name` returns a vector of double values. If the text file contains two values per line, vfile returns a vector of complex values. `name` may start with a symbolic directory name (template $XXX). See the following section for more information. Remember to add a backslash after the symbolic directory name.

SYMBOLIC DIRECTORY NAMES

You can use the following symbolic directory names with vfile `name` and data_file `name` functions.

Name	Description	Typical directory
`$MWO`	Replaced with the AWR Design Environment platform installation folder name.	C:\Program Files (x86)\AWR\AWRDE_version_
`$PRJ`	Replaced with the name of the folder containing the project file.	C:\Users_user_\Documents_project_folder_name_
`$APPDATAUSER`	Replaced with the name pointing to the AppDataUser directory.	C:\Users\user\AppData\Local\AWR\Design Environment_version_
`$DATA`	Replaced with the name of the Data folder within the `\AWRDE` installation folder	C:\Program Files (x86)\AWR\AWRDE_version_\Data

COMPLEX NUMBER FUNCTIONS

You can use the following complex number functions, where `z` is a complex number.

Function	Description
`real(z)`	Returns the real part of a complex number.
`imag(z)`	Returns the imaginary part of a complex number.
`angle(z)`	Returns the angle (in radians) of a complex number.
`conj(z)`	Returns the complex conjugate of a complex number.
`abs(z)`	Returns the magnitude of a complex number.

SWEPT FUNCTIONS

You can use the following functions for generating a vector of real values. These functions are useful for generating a frequency sweep in the Microwave Office program for use with the SWPFRQ and SWPVAR blocks.

Function	Description
`concat(a, b)`	Concatenates vectors `a` and `b`. Arguments must have the same type.
`points(start, stop, points)`	Generates a vector of real numbers from `start` to `stop`, with a total length of `points`.

Function	Description
`stepped(start, stop, step)`	Generates a vector of real numbers from `start` to `stop` with `step` size between consecutive numbers.
`swpdec(start, stop, points)`	Generates a vector of real numbers from `start` to `stop` with `points` per decade.
`swpoct(start, stop, points)`	Generates a vector of real numbers from `start` to `stop` with `points` per octave.
`swplin(start, stop, points)`	Generates a vector of real numbers from `start` to `stop` with a total length of `points`.
`swpspan(center, span, points)`	Generates a vector of real numbers with a `center` and a `span` with a total length of `points`.
`swpspanst(center, span, step)`	Generates a vector of real numbers with a `center` and a `span` with `step` size between consecutive numbers.
`swpstp(start, stop, step)`	Generates a vector of real numbers from `start` to `stop` with `step` size between consecutive numbers.

FUNCTIONS THAT REFERENCE GENERALIZED MDIF DOCUMENTS

The following functions support pulling values from generalized MDIF data documents. You can use these functions in the Global Equations or within a schematic document. You cannot use them directly in Output Equations, however since Output Equations can reference Global Equations, you can still use the values returned from these functions from within the Output Equations.

- IndexMD: This function is used to retrieve values by index. This function takes a variable number of arguments and returns a scalar, 1-D or 2-D array.

  IndexMD("MDIF_DOC_NAME", "IndepVar1_Name", Var1_Index,... "IndepVarNS_Name", VarNS_Index, "DependVar_Name")

Argument	Description
`"MDIF_DOC_NAME"`	Document name for the MDIF is the name of the document in the Project Browser (not the file name)
`"IndepVar1_Name"`	Name of the independent variable used to define a slice of data. Must match one of the independent variable names within the MDIF.
`Var1_Index`	Index of the "IndepVar1_Name" variable used to define the slice of data.
`"IndepVarNS_Name"`	Name of the independent variable used to define a slice of data. Must match one of the independent variable names within the MDIF. 'NS' defines how many slice dimensions are passed into the function and must be no more than the number of dimensions in the MDIF.
`VarNS_Index`	Index of the "IndepVarN_Name" independent variable used to define the slice of data.
`"DependVar_Name"`	Defines the dependent value quantity returned from the function. Name must match one of the dependent variable names within the MDIF. Return type is either real or complex, depending on type of the named variable within the MDIF file. Dimensions of the returned data are ND-NS, where ND is the number of dimensions of the data in the MDIF file, and NS is the number of slices defined as arguments to the function. A

Argument	Description
	maximum of two dimensions can be returned, so function returns an error if ND-NS > 2. When ND=NS, a scalar is returned (all dimensions are specified).

- LookupMD: This function is very similar to IndexMD, except the values of the independent variables are passed in instead of the indices. This function does not do any interpolation, so it returns the closest match for the passed in values.

  LookupMD("MDIF_DOC_NAME", "IndepVar1_Name", Var1_Value,... "IndepVarNS_Name", VarNS_Value, "DependVar_Name")

- InterpMD: This function can be used to interpolate values from the arbitrary dimension data set. Even though the syntax looks similar to IndexMD or LookupMD, the usage is more complex, partly because the variables used to define the slice dimensions can be either independent or dependent values. This makes the function very flexible, although it should be used with care. The main reason for allowing dependent variables to be used for slice dimensions is for the case where a sweep is represented as an integer index within the independent values in the MDIF, but the actual value of the swept quantity is stored as a dependent value. For example, for load pull files, the power may be swept by an iPower index, and then you can use different dependent power values (such as Pava_Src) as the dimension to interpolate over).

  InterpMD("MDIF_DOC_NAME", "Var1_Name", Var1_Values,... "VarND_Name", VarND_Values, "DependVar_Name")

Argument	Description
`"MDIF_DOC_NAME"`	Document name for the MDIF is the name of the document in the Project Browser (not the file name)
`"Var1_Name"`	Name of the variable used to define a slice of data. Must match one of the independent or dependent variable names within the MDIF.
`Var1_Values`	Values for the variable that defines the slice of data. Value can be a scalar or a vector. If vector, the length of the vector determines one of the dimensions of the value returned from the function. If MDIF data is on a regular grid and the specified value matches one of the values on that regular grid, this function automatically reduces the dimension of the space being interpolated over. This can be useful for performance reasons, and it can also be used to get around the limits on the interpolation dimensions.
`"IndepVarND_Name"`	Name of the variable used to define a slice of data. Must match one of the independent or dependent variable names within the MDIF. Number of slices passed in must be equal to the number of dimensions in the MDIF file (ND=NS).
`VarND_Values`	Index of the "IndepVarN_Name" independent variable used to define the slice of data.
`"DependVar_Name"`	Defines the dependent value quantity returned from the function. Name must match one of the dependent variable names within the MDIF. Return type is either real or complex, depending on type of the named variable within the MDIF file. Dimensions of the returned data are a scalar if all the Var[X]_Values are scalar. If one of the Var[X]_Values is a vector, the returned value has the same dimension as Var[X]_Values. If two arguments (Var[X]_Values and Var[Y]_Values) have vector values, the returned value has a dimension equal to Size[X]*Size[Y]. A maximum of 2 dimensions can be returned, so the function returns an error if more than two Var[X]_Values are vectors.

NOTES ON INTERPOLATION

- The type and limitations of the interpolation depend on the structure of the data. If the data is on a regular grid and all data points are defined on that grid, then linear interpolation is used, which can support up to 10 dimensions of data. For data that does not conform to a regular grid, thin plate spline interpolation is used, which can currently interpolate up to 3 dimensions. It is also possible to have mixed regular and irregular data (when you interpolate over dependent values from a data set that has regular independent values). In these cases, you can reduce the number of dimensions interpolated over by making sure one or more of the Var[X]_Values are scalar values that correspond to a matching independent value in the MDIF data set. For each matching value, the number of dimensions required to be interpolated is reduced by one.

INTERPOLATION OPTIONS
- The MDIF data document has interpolation options you can control (choose **Options > Project Options** and click the **Interpolation/Passivity** tab on the Project Options dialog box). Not all options that you can set are used for MDIF data interpolation. Currently, only the **Method** option changes the interpolation, and can be **Linear**, **Rational function**, or **Spline curve**. When the interpolation is performed over 1-dimension, these settings are always respected. Interpolation over a higher than 1-dimension data set is still a 1-d interpolation under certain conditions (see the previous paragraph). The following table shows what interpolation methods you can use with what type of data. If an incompatible method is selected, a compatible method is auto-selected instead. For most use, the **Linear** default is good, otherwise the code chooses another method when needed.

Structure	Dimensions	Methods Supported
Uniform	1	Linear, Rational, Spline
Uniform	2-3	Linear, Spline
Uniform	3-10	Linear
Non-uniform	1	Linear, Rational, Spline
Non-uniform	2-3	Spline
Non-uniform	>3	Not currently supported

You can use the following built-in variables in the Microwave Office program to return simulation frequencies and temperatures.

BUILT-IN VARIABLES IN Microwave Office

Variable	Description
`_ANG_U`	Reserved for Microwave Office use.
`_CAP_U`	Reserved for Microwave Office use.
`_COND_U`	Reserved for Microwave Office use.
`_CURR_U`	Reserved for Microwave Office use.
`_FREQ`	Variable containing the simulation frequency (in Hz) when used in a schematic. For HB analysis, _FREQ contains the frequency set (fundamental plus harmonics and products) for the current sweep point. _FREQ contains the project frequency list when used in the Output Equation window. Units are always in Hz. Reserved for Microwave Office use.
`_FREQH1`	Variable containing the first tone of a harmonic balance simulation. Reserved for Microwave Office use, in the schematic window.[a]
`_FREQH2`	Variable containing the second tone of a harmonic balance simulation. Reserved for Microwave Office use, in the schematic window.[a]

Variable	Description
_FREQH3	Variable containing the third tone of a harmonic balance simulation. Reserved for Microwave Office use, in the schematic window.[a]
_FREQ_U	Reserved for Microwave Office use.
_IND_U	Reserved for Microwave Office use.
_LEN_U	Reserved for Microwave Office use.
_PI	Reserved for Microwave Office use.
_RES_U	Reserved for Microwave Office use.
_TEMP	Variable used for the default temperature of many models. The units for this variable are determined by the project temperature units ("Project Options Dialog Box: Global Units Tab "), unless **Dependent parameters use base units** is selected on the Options dialog box **Schematic** tab, in which case the variable has units of Kelvin. [b]
_TEMPK	Variable used to set the noise temperature (in Kelvin) for any model that does not have a temperature parameter. Passive elements have their noise contributors scaled by _TEMPK/290.0.[b] Reserved for Microwave Office use.
_TIME_U	Reserved for Microwave Office use.
_VOLT_U	Reserved for Microwave Office use.

[a]Units are always in Project Units unless **Dependent parameters use base units** is selected for the project. See "Determining Project Units" for details.

[b]See "Using Temperature in Simulations" for information on the correct use of `_TEMP` and `_TEMPK`.

BUILT-IN VARIABLES IN VSS

You can use the following built-in variables in the VSS™ (VSS) system diagram windows:

Variable	Description
_BLKSZ	Variable containing the **Block Size** setting from the System Simulator Options dialog box **Advanced** tab.
_DRATE	Variable containing the default data rate computed from: _DRATE = _SMPFRQ/_SMPSYM when _SMPFRQ is in Hz. The units are cycles per second. This variable may be swept, although if so, _SMPFRQ, _SMPSYM, _TFRAME, and _TSTEP are not updated.
_FREQ	For RF Budget Analysis, array variable containing the frequency list from the System Simulator Options dialog box **RF Frequencies** tab. For RF Inspector and Time Domain simulations, variable containing the first frequency from the frequency list in this dialog box. Units are always in Hz. Reserved for Microwave Office use.
_SMPFRQ	Variable containing the default **Sampling Frequency Span** from the System Simulator Options dialog box **Basic** tab. _SMPFRQ = _DRATE*_SMPSYM when _SMPFRQ is in Hz. The units are in Hertz if system diagrams are configured to use base units for dependent parameters on the Project Options dialog box **Schematics/Diagrams** tab. If they are not, the units are in the project frequency units. This variable may be swept, although if so, _DRATE, _SMPSYM, _TFRAME, and _TSTEP are not updated.
_SMPSYM	Variable containing the default **Oversampling Rate** from the System Simulator Options dialog box **Basic** tab. _SMPSYM = _SMPFRQ/_DRATE when _SMPFRQ is in Hz.

Variable	Description
_TAMB	Variable containing the **Ambient Temperature** from the System Simulator Options dialog box **RF Options** tab. The units are in Kelvin if system diagrams are configured to use base units for dependent parameters on the Project Options dialog box **Schematics/Diagrams** tab. If they are not, the units are in the project temperature units. This variable can be swept.
_TFRAME	Variable containing the default time step between data samples, _TFRAME = 1/_DRATE.
_TSTEP	Variable containing the default time step between waveform samples, _TSTEP=1/_SMPFRQ when _SMPFRQ is in Hz.
_Z0	Variable containing the impedance setting from the System Simulator Options dialog box **RF Options** tab, **Impedance** option.

USING BUILT-IN FREQUENCY VARIABLES

The AWR Design Environment platform allows you to sweep frequencies in several different ways, and each measurement can use a different frequency sweep. When a circuit is analyzed, several different frequency sweeps may be performed to complete all of the measurements. Built-in variables are available that represent the frequency during each simulation.

_FREQ is the reserved variable name for the simulation frequency vector (set of values). For linear simulations, _FREQ is the set of frequencies in the sweep. For nonlinear simulations, _FREQ is the set of harmonic (spectral) frequencies being analyzed. For example, if you make a resistor a function of _FREQ, it has a list of values: one for each value in _FREQ. For linear analysis, each value is a function of the corresponding sweep frequency. However, for each frequency point in a nonlinear simulation, that same resistor has a different list of values, each of which is a function of the corresponding harmonic frequency in the analysis, so the resistor is properly frequency-dependent in both cases.

You can access the nonlinear input (tone 1) frequency sweep with the _FREQH1 variable. This allows you to make the frequency of one tone dependent on another. For example, assume you use a PORT1 element as the tone 1 input signal, and want a second tone source with a frequency 0.01 less than tone 1. You can place a PORTFN element with the parameters Tone=2 and Freq=_FREQH1-0.01. (The tone number used by a signal port is either set in its parameter list, or on the **Port** tab of the Element Options: PORT dialog box (right-click the port and choose **Properties**).

With one PORT1 element (using tone 1) as the input signal, harmonic balance settings that specify 5 harmonics for tone 1, and a frequency sweep of 1, 2, and 3 GHz, assuming the project global units are set to GHz, during the sweep, the following are the values of the variables for each frequency in the sweep:

_FREQH1=1 _FREQ={1e9,2e9,3e9,4e9,5e9}

_FREQH1=2 _FREQ={2e9,4e9,6e9,8e9,10e9}

_FREQH1=3 _FREQ={3e9,6e9,9e9,12e9,15e9}

Note that _FREQ values are always in Hz. Equations that use _FREQ to calculate frequency-dependent element and parameter values should anticipate this.

GLOBAL CONSTANTS

You can use the following global constants without using the constants command:

Global Constant	Description
_PI	The mathematical constant π (3.14159...)

Global Constant	Description
`i, j`	The imaginary number defined by √-1.

You can use the following global constants by using the constants("name") function. For example, call constants("boltzmann") to get the value 1.3806226e-23.

Global Constant	Value
`e`	2.7182818284590452354
`ln10`	2.30258509299404568402
`pi`	3.14159265358979323846
`c0`	2.997924562e8
`e0`	8.85418792394420013968e-12
`u0`	3.14159265358979323846*4e-7
`boltzmann`	1.3806226e-23
`qelectron`	1.6021918e-19
`planck`	6.6260755e-34
`M_E`	2.7182818284590452354
`M_LOG2E`	1.4426950408889634074
`M_LOG10E`	0.43429448190325182765
`M_LN2`	0.69314718055994530942
`M_LN10`	2.30258509299404568402
`M_PI`	3.14159265358979323846
`M_TWO_PI`	6.28318530717958647652
`M_PI_2`	1.57079632679489661923
`M_PI_4`	0.78539816339744830962
`M_1_PI`	0.31830988618379067154
`M_2_PI`	0.63661977236758134308
`M_2_SQRTPI`	1.12837916709551257390
`M_SQRT2`	1.41421356237309504880
`M_SQRT1_2`	0.70710678118654752440
`M_DEGPERRAD`	57.2957795130823208772
`P_Q`	1.6021918e-19
`P_C`	2.997924562e8
`P_K`	1.3806226e-23
`P_H`	6.6260755e-34
`P_EPS0`	8.85418792394420013968e-12
`P_U0`	3.14159265358979323846*4e-7
`P_CELSIUS0`	273.15

12.7.4. Using String Type Variables

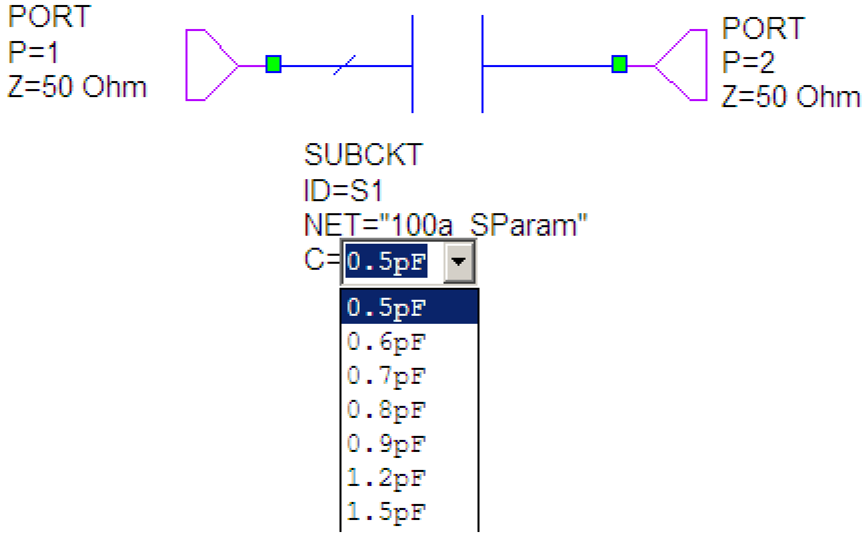
Equations can use string-type variables in addition to real and complex types. You must manually add the quotes to differentiate string-type data from string-type equations.

You must enclose the NET parameter for subcircuit elements in quotes. For example:

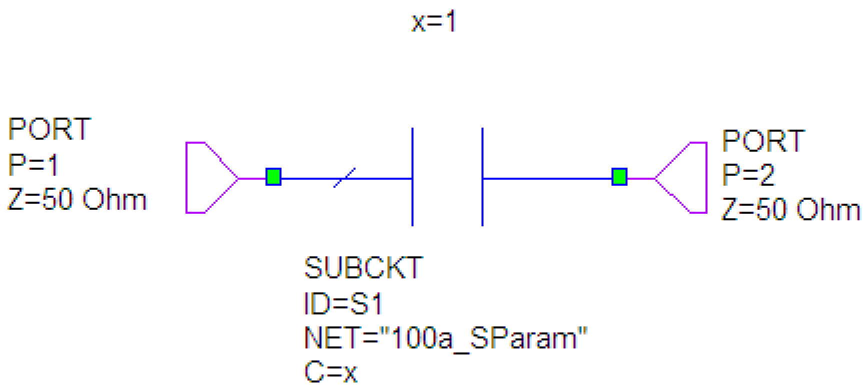
```
NET="One"
```

You can concatenate strings using the '+' operator.

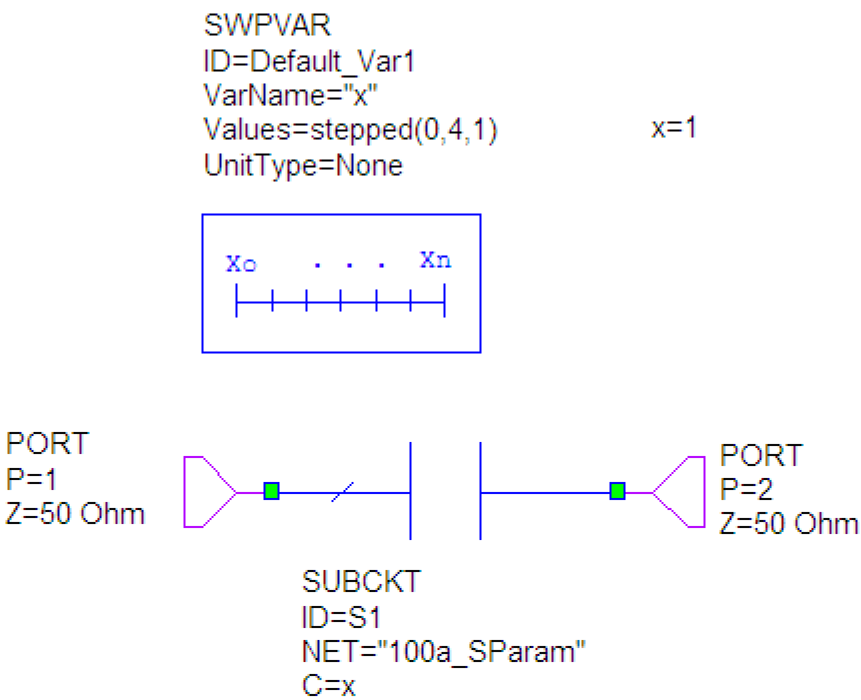
Some model parameters are a list of available values. These are different than string-types; you can tell because there is no quote around the string name. An excellent example is an MDIF file. In the following figure, note the NET parameter is set as a string-type by the quotes, but the C parameter has a list of available settings.



For any model parameter of this type, you can type in a variable name for the parameter, then set the variable to an integer that indicates the position in the drop-down list to use, starting with 0. You do NOT use the text available in the drop-down menu. The following example shows the C parameter set to a variable, and the variable set to 1, so it uses the second item in the list.



If you want to sweep through the first five elements of this model, you set up the schematic accordingly.



12.7.5. Defining Vector Quantities

Equations support real, complex, or string type vector quantities. To define a vector, use the following syntax:

```
x={10, 25, 30, 50}
x={i, 3*i, 2*i}
x={"One", "Two", "Three", "Four"}
```

Vectors cannot mix types, such as strings and numbers. The vector type is determined by the last member. To reference a particular value, use the following syntax:

```
x={10, 25, 30, 50}  x[1]:
10
x={i, 3*i, 2*i}   x[2]:
(0,3)
x={"One","Two","Three","Four"} x[3]: "Three"
```

The array index must be in the range `[1,N]`, where `N` is the number of items in the vector. You can also define a vector using the `stepped` function with the following syntax:

```
stepped(start,stop,step)
```

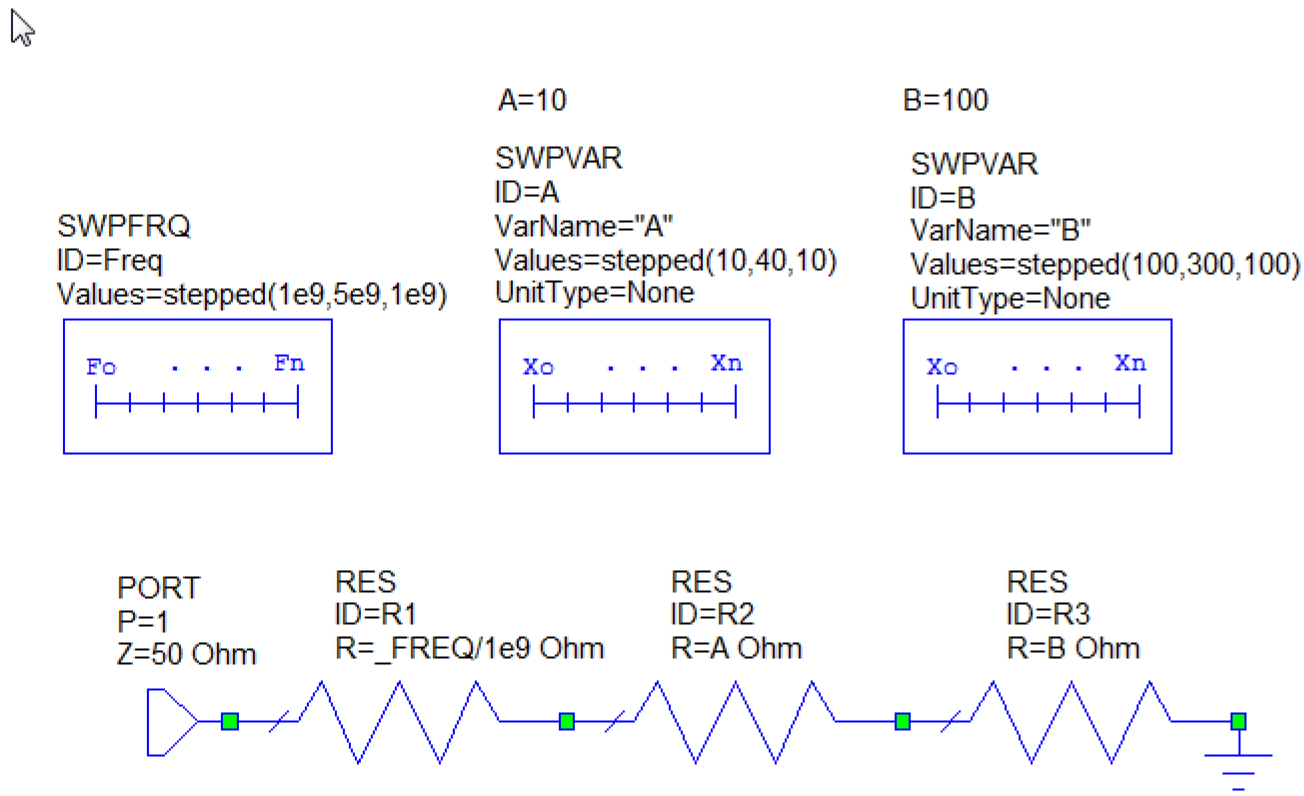
If `start` is less than `stop`, then `step` must be greater than 0. If `stop` is less than `start`, then `step` must be less than 0.

Array references are independent variables that you can tune and optimize, and which are always constrained between 1 and N. You can override these constraints to fall within this range. Optimizers must support discrete optimization of vector quantities, except for gradient-based optimizers, which cannot function with discrete values. AWR Design Environment platform Pointer and Random optimizers support discrete values.

12.7.6. Swept Measurement Data in Output Equations

The example presented here demonstrates swept measurement data in an output equation.

The schematic in the following figure generates a three dimensional (3D) sweep. Zin measured at port 1 is the sum of the simulation frequency in GHz and variables A and B.



You can consider the Zin data as being stored in the following format.

B=100

Freq	A=10	A=20	A=30	A=40
1e9	111	121	131	141
2e9	112	122	132	142
3e9	113	123	133	143
4e9	114	124		
5e9	115	125		

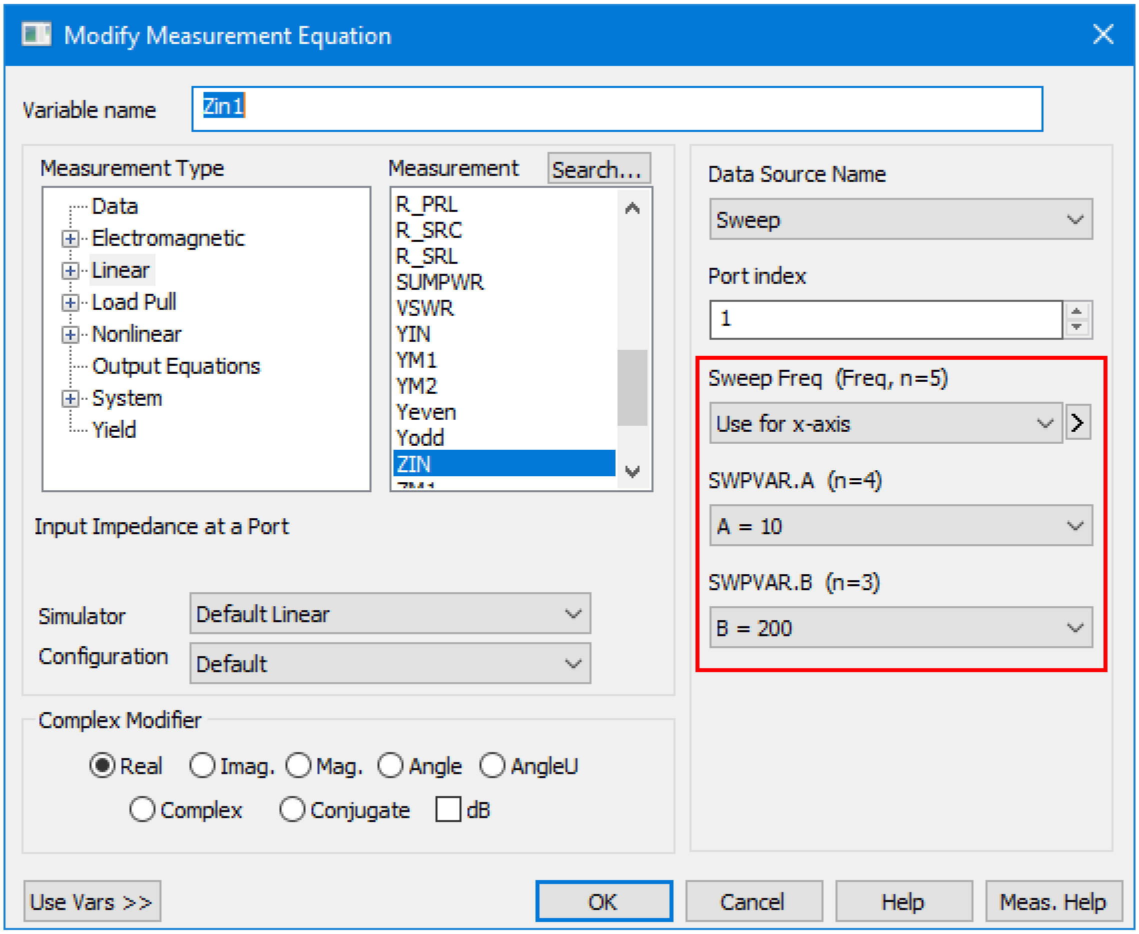
B=200

Freq	A=10	A=20	A=30	A=40
1e9	211	221	231	241
2e9	212	222	232	242
3e9	213	223		
4e9	214	224		
5e9	215	225		

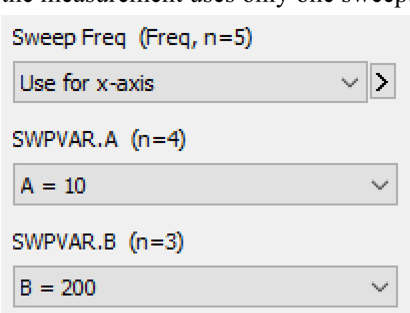
B=300

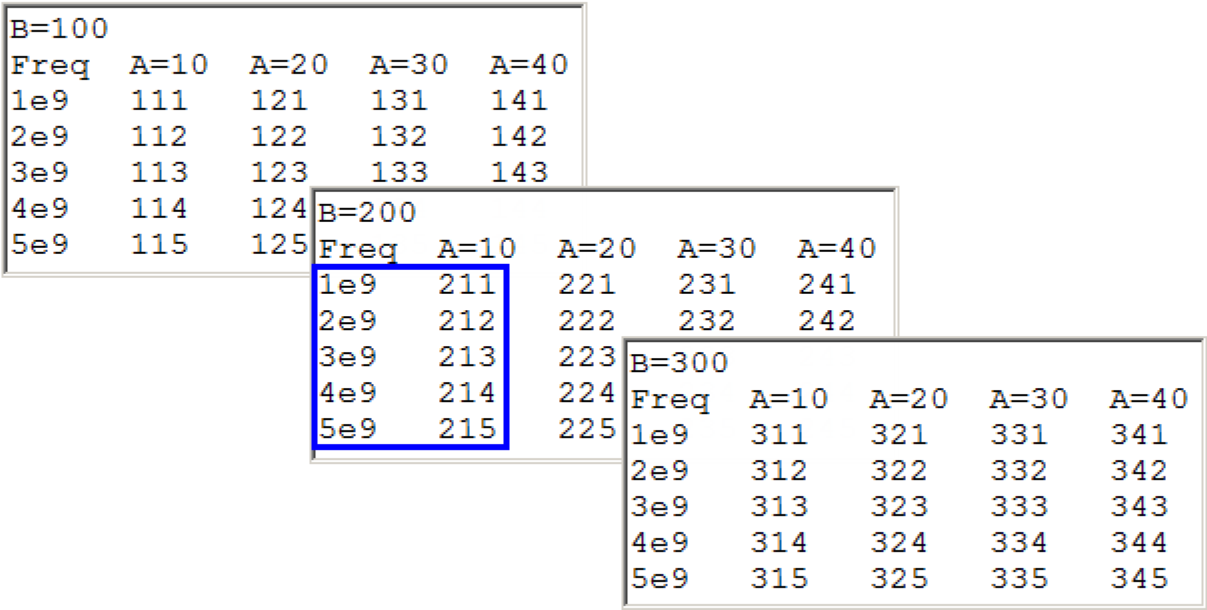
Freq	A=10	A=20	A=30	A=40
1e9	311	321	331	341
2e9	312	322	332	342
3e9	313	323	333	343
4e9	314	324	334	344
5e9	315	325	335	345

When you specify a measurement, the sweep settings in the Modify Measurement Equation dialog box allow a form of filtering that can take “slices” from the data.



The following examples show the selected options from the Modify Measurement Equation dialog box, the portions of the data that are extracted, and the resulting data when the measurement is used in an Output Equations document. Each example increments an integer suffix on the variable name to distinguish it from the others. The simplest case is when the measurement uses only one sweep. Here, frequency is specified as the x-axis, but A and B are fixed.





B=100

Freq	A=10	A=20	A=30	A=40
1e9	111	121	131	141
2e9	112	122	132	142
3e9	113	123	133	143
4e9	114	124		
5e9	115	125		

B=200

Freq	A=10	A=20	A=30	A=40
1e9	211	221	231	241
2e9	212	222	232	242
3e9	213	223		
4e9	214	224		
5e9	215	225		

B=300

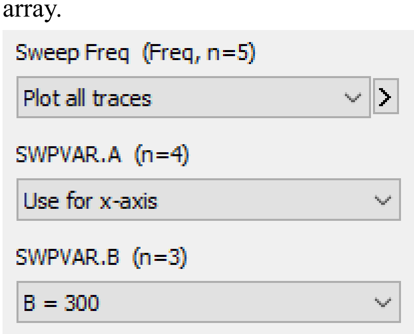
Freq	A=10	A=20	A=30	A=40
1e9	311	321	331	341
2e9	312	322	332	342
3e9	313	323	333	343
4e9	314	324	334	344
5e9	315	325	335	345

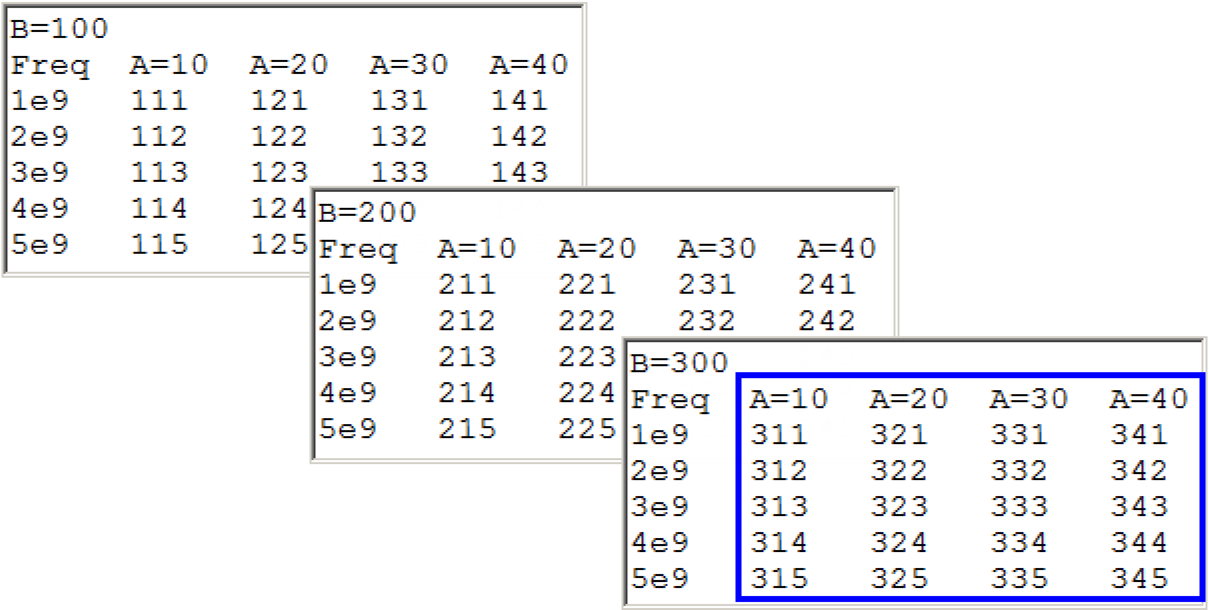
In output equations:

```
Zin1 = Sweep.$Freq:Re(ZIN(1))[X,1,2]
Zin1:  {111,112,113,114,115}
swpvals(Zin1): {1e9,2e9,3e9,4e9,5e9}
```

Note that the SWPFRQ block is used with ID=Freq as the sweep frequencies of the measurement defining Zin1. Other than the variable name used, all of the information in output equations is numeric. Here, the variable Zin1 becomes the 1D array (vector) of Zin values when: frequency is swept, A=10, and B=200. The swpvals( ) function can get the vector of corresponding frequency values. However, the names of the swept variables do not mean anything in output equations. If the swept variable A is set to **Select with tuner**, the size of the vectors stays the same, but the Zin1 values correspond to the column selected by tuning A.

The next example shows that when one of the swept variables is set to **Plot all traces**, the output equation becomes a 2D array.





The following is a visual representation of the data in the output equation:

```
swpvals(Zin2):          Zin2:
        10        311,312,313,314,315
        20        321,322,323,324,325
        30        331,332,333,334,335
        40        341,342,343,344,345
```

Because the swept variable A is set to **Use for x-axis**, its values are now the sweep values. The data now has as many columns as there are frequencies, and one row for each A value. To see how the data is stored, plot the measurement in a tabular graph. The first column of the tabular graph shows the sweep values. The remaining columns show the measurement data, with rows corresponding to the sweep values in order, and the measurement values for the nth parameter value in the nth column after the sweep values.

In output equations:

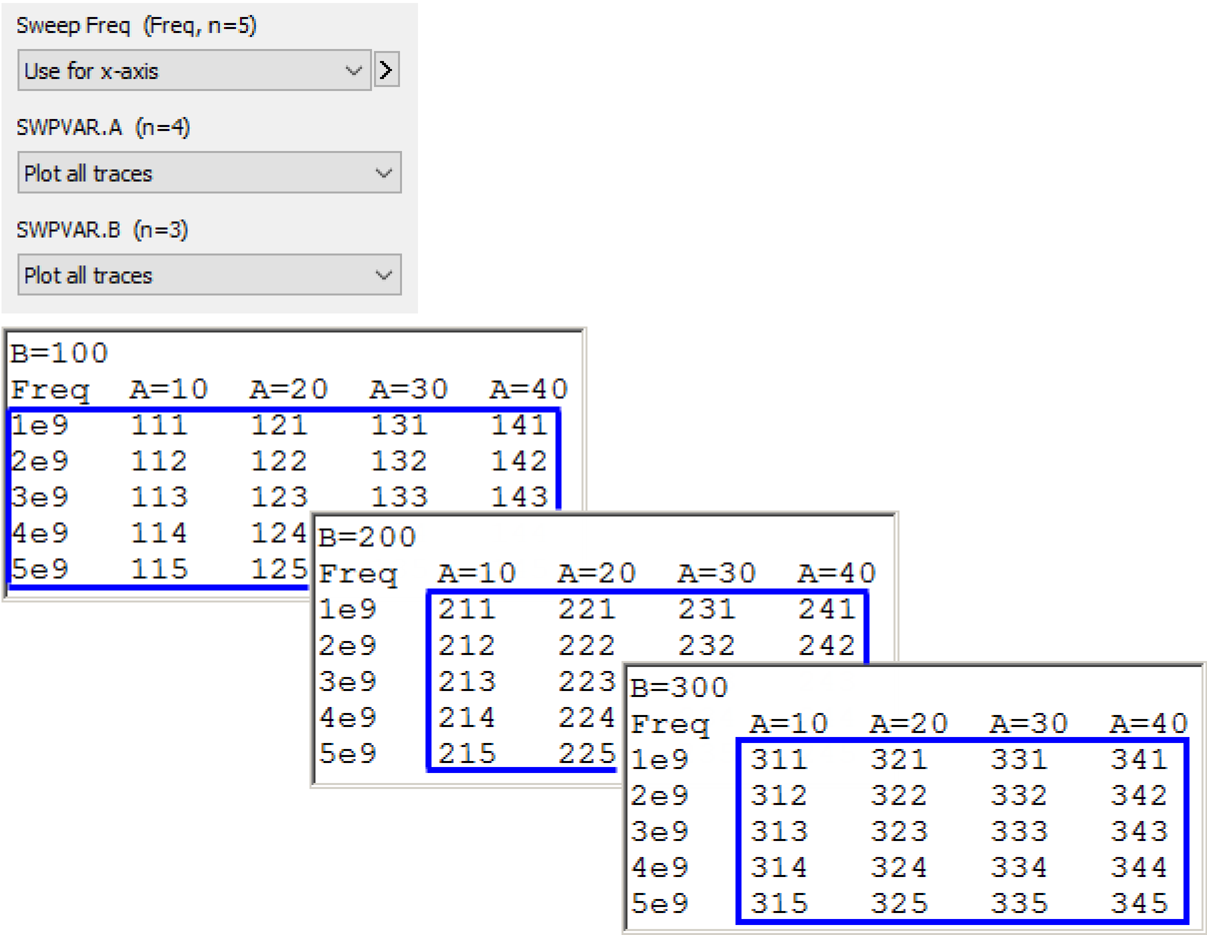
```
Zin2 = Sweep.$Freq:Re(ZIN(1))[*,X,3]
asize(Zin2):{4,5}
Zin2[*,3]: {313,323,333,343}     Values for 3rd frequency point
Zin2[2,*]: {321,322,323,324,325} Values for 2nd sweep point of A
```

If there are other equations that calculate a variable “x”, and you want the row of Zin2 data for the A value closest to x, plotted vs. frequency, you can use the following equations:

```
A_index = Findindex(swpvals(Zin2),x)
Zin2_row = Zin2[A_index,*]
Answer = plot_vs2(Zin2_row,swpvals(Zin1),1)
```

You can use multiple output equations for the same measurement, but with a different swept parameter set to **Use for x-axis**, in order to access the list of values for that parameter. In this example, Zin2 does not include the frequency values; its sweep values are the values of the swept variable A. The equation for Answer refers to the output equation in the previous example: swpvals(Zin1) provides the frequency list.

In the following example, swept variable B is also set to **Plot all traces**. When additional swept parameters are set to **Plot all traces**, the output equation data is still a 2D array, with a column for each combination of parameter values.



The data displays similar to the following (as in a tabular graph):

```
swpvals(Zin3):                         Zin3:
        1e9     111,121,131,141,  211,221,231,241,  311,321,331,341
        2e9     112,122,132,142,  212,222,232,242,  312,322,332,342
        3e9     113,123,133,143,  213,223,233,243,  313,323,333,343
        4e9     114,124,134,144,  214,224,234,244,  314,324,334,344
        5e9     115,125,135,145,  215,225,235,245,  315,325,335,345
```

The first four columns of Zin3 data correspond to the four values of A in increasing order, and the lowest value of B. The next four columns are for the next B value, and so on. The order of sweeps is alphabetical, based on the ID parameter of the SWPVAR block, except that frequency is always the first (fastest) sweep when set to **Plot all traces**. (Spaces are added for clarity; the data does not distinguish columns other than by index number.)

In output equations:

```
Zin3 = Sweep.$Freq:Re(ZIN(1))
asize(Zin3):{5,12} dim=asize(Zin3)  rows=dim[1]=5 cols=dim[2]=12
```

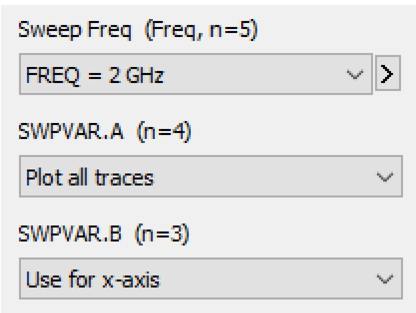
The function asize(Zin3) returns the dimensions of the array as {5,12} (5 rows, and 12 columns). However, without additional information, there is no way to determine how many swept variables there are, or how many values each has. Again, you can use the other equations to determine the number of A values:

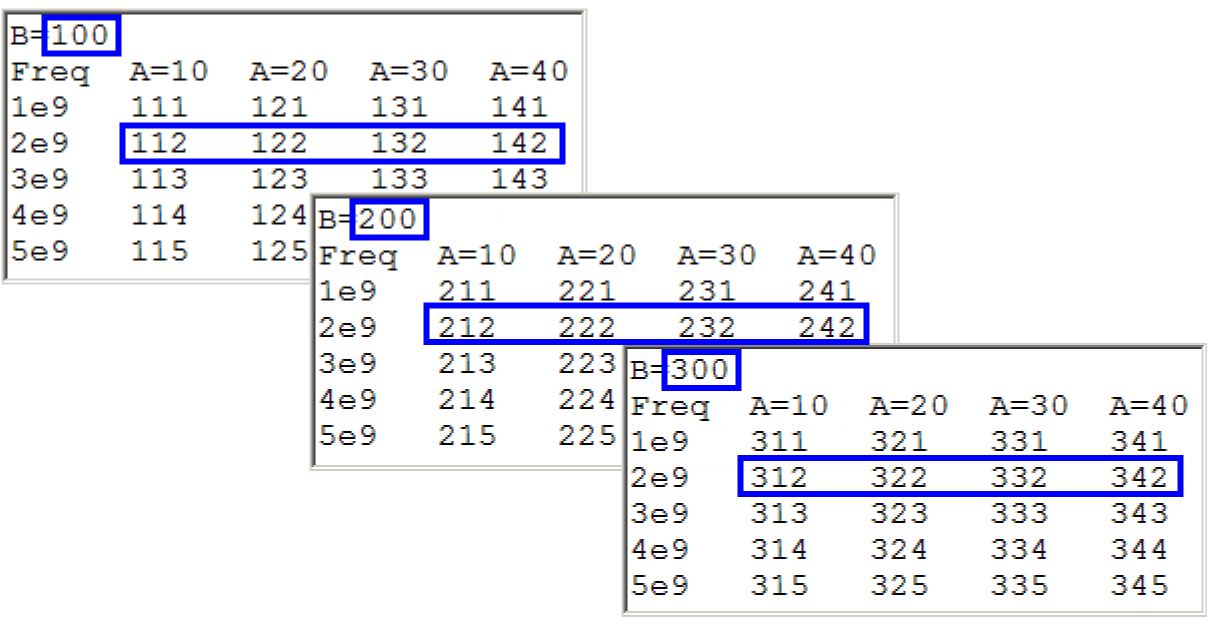
```
Num_A = asize(swpvals(Zin2)) Num_A:4
```

This shows that every 4th column in Zin3 data corresponds to the same A value. (You use equations to determine this, rather than setting Num_A=4, so that everything scales automatically if the schematic is edited and the number of sweep points changes.) To get the Zin values for the 4th frequency, and the A value closest to x, at all B values:

```
Zin3[3,stepped(A_index,cols,Num_A)]: {134,234,334}
```

To plot this versus values of B, you need another output equation where the measurement sets B to **Use for x-axis**, so you can get its sweep values as follows:





The data displays as follows:

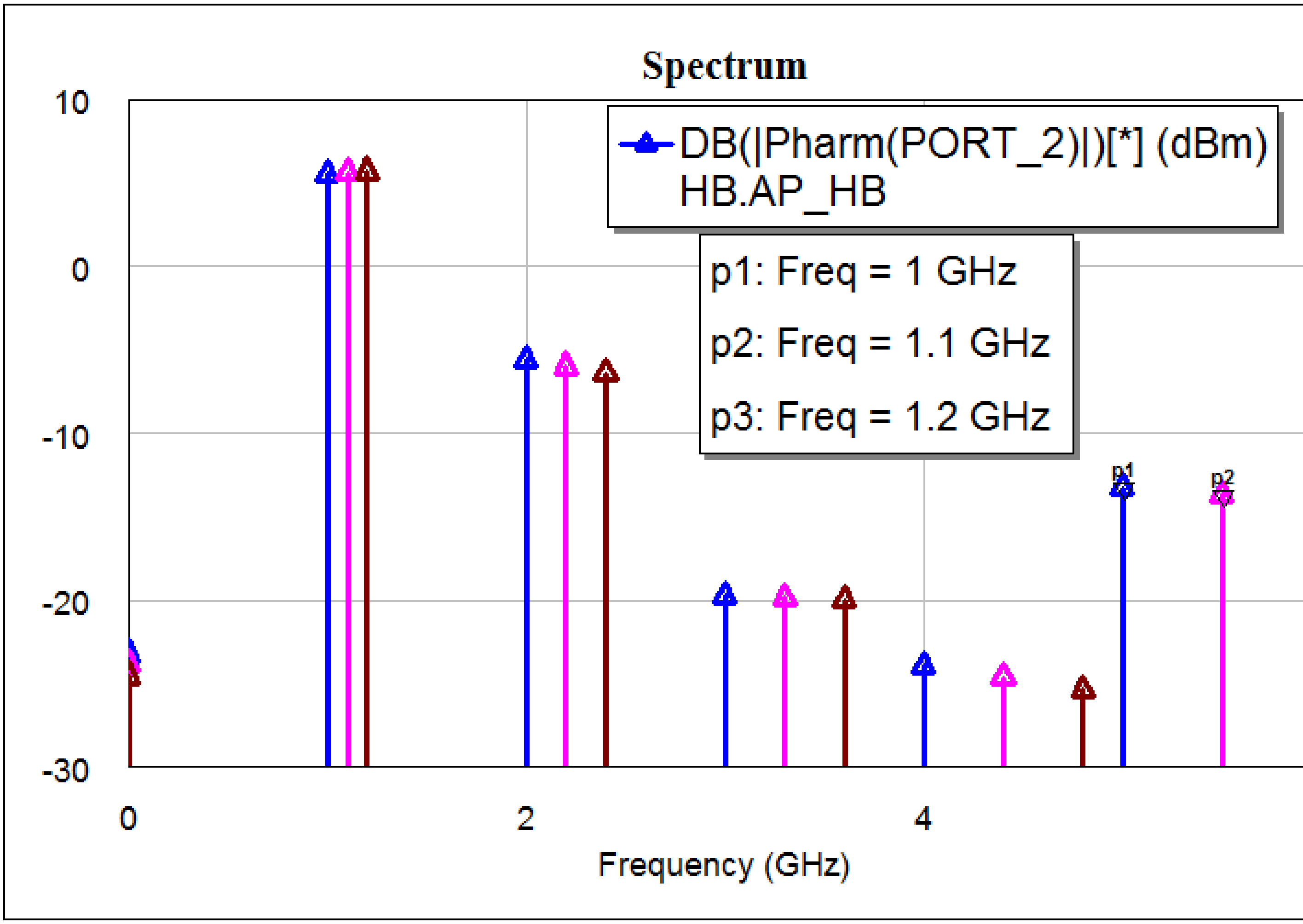
```
swpvals(Zin4):  Zin4:
         100    112,122,132,142
         200    212,222,232,242
         300    312,322,332,342
```

In output equations:

```
Zin4 = Sweep.$Freq:Re(ZIN(1))[2,*,X]
Zin3_vs_B = plot_vs2(Zin3,swpvals(Zin4),4)
```

12.7.6.1. Inconsistent X-axis Values

Some measurements have a different list of x-axis values for each simulation in a sweep. The simplest example of this is a frequency spectrum measurement like Pharm, after a harmonic balance simulation where the input frequency is swept. For example, if the input frequency is swept through the values 1, 1.1, and 1.2 (GHz), and the harmonic balance simulation is set up to simulate 5 harmonics, then the Pharm measurement consists of 6 points per input frequency (DC and 5 harmonics). However, at each input frequency, F0, there is a different set of harmonic frequencies for the x-axis, N*F0, where N=1,2,3,4,5. If you set the frequency sweep to **Plot all traces**, and plot the measurement on a rectangular graph, it displays as shown in the following figure (traces are set to **Step Color**, so each color represents a different input frequency):



If you use this measurement in output equations:

```
HB = HB.AP_HB:DB(|Pharm(PORT_2)|)[*]
```

The data displays as follows:

```
swpvals(HB):                              HB:
        1        0       -23.227    0        -23.867    0         -24.546
        2        1e9      5.5535    1.1e9     5.6228    1.2e9      5.6869
        3        2e9     -5.5815    2.2e9    -6.0023    2.4e9     -6.4105
        4        3e9     -19.718    3.3e9    -19.822    3.6e9     -19.979
        5        4e9     -23.948    4.4e9    -24.617    4.8e9     -25.334
        6        5e9     -13.251    5.5e9    -13.67     6e9       -14.158
```

The vector returned by swpvals cannot hold all of the x-axis values; so the function just returns a vector of integers (row index numbers). Instead, the data in the array "HB" holds one pair of columns, x and y values, for each input frequency: odd columns are x-axis (spectral frequency) values, and even columns are the corresponding harmonic power levels. There are simple ways of segregating the x and y values if necessary. For example:

```
asize(HB):{6,6} dim=asize(HB)  rows=dim[1]=6 cols=dim[2]=6

Frequencies = HB[*,stepped(1,cols-1,2)]   ; all rows, but odd columns only

Frequencies: { {0,0,0},{1,1.1,1.2},{2,2.2,2.4},{3,3.3,3.6},…}
```

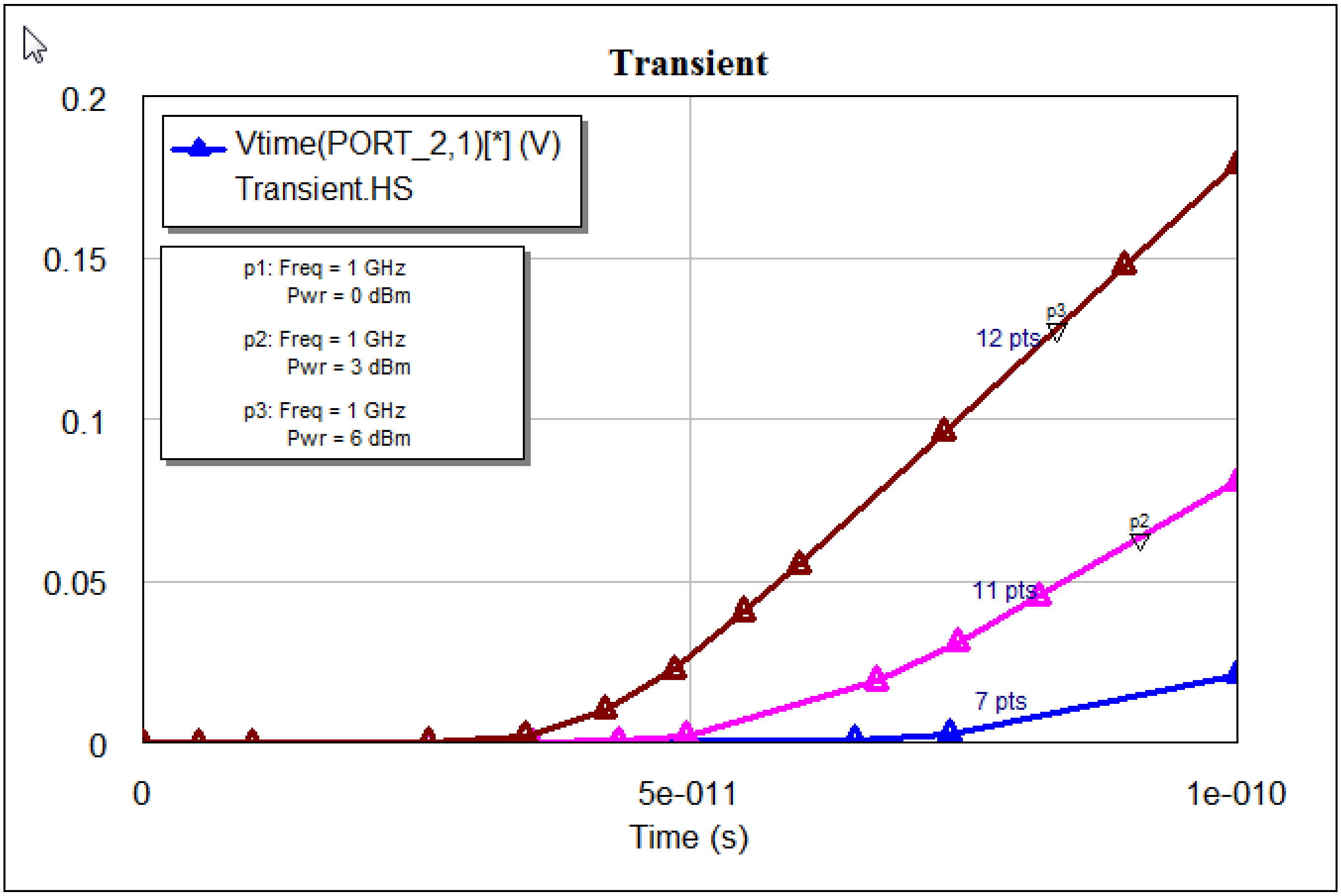
or, visually:

```
        0      0        0
        1e9    1.1e9    1.2e9
        2e9    2.2e9    2.4e9
        3e9    3.3e9    3.6e9
        4e9    4.4e9    4.8e9
        5e9    5.5e9    6e9
```

If an additional parameter (for example, the input power level) is swept, another six columns of data are added for each additional input power level.

12.7.6.2. Inconsistent Number of Points in Each Sweep

Some measurements do not have an identical number of points in each sweep. For example, transient analyses have variable time step sizes, and may require a different number of time points when simulating each value of a swept parameter. A Vtime measurement from a transient simulator may have a different number of points for each simulation in a sweep.



In output equations:

```
Tran = Transient.HS:Vtime(PORT_2,1)[*]
```

Here, the size of the data array depends on the sweep with the largest number of points, and the “empty” elements of the array are set to the highest number possible. Equations recognize and ignore these values: math operations are not performed on them. These values display in brackets in the following table (rounded to 4 digits).

0	5.8921e-008	0	5.8921e-008	0	5.8921e-008
5e-012	1.2765e-007	5e-012	1.7561e-007	5e-012	2.7556e-007
1e-011	2.7636e-007	1e-011	5.2287e-007	1e-011	1.2869e-006
2.6e-011	3.2271e-006	2.6e-011	1.6813e-005	2.6e-011	0.00017125
6.4952e-011	0.00095671	3.5631e-011	0.00013135	3.4824e-011	0.0021303
7.3714e-011	0.0028257	4.3336e-011	0.00064518	4.2103e-011	0.010124
1e-010	0.021405	4.95e-011	0.0020934	4.85e-011	0.022758
[1.798e308]	[1.798e308]	6.6888e-011	0.019187	5.4897e-011	0.040639
[1.798e308]	[1.798e308]	7.434e-011	0.030976	5.9902e-011	0.055256
[1.798e308]	[1.798e308]	8.1792e-011	0.045444	7.3073e-011	0.095886
[1.798e308]	[1.798e308]	1e-010	0.08099	8.9661e-011	0.14767
[1.798e308]	[1.798e308]	[1.798e308]	[1.798e308]	1e-010	0.17911

Chapter 13. Wizards

The Cadence® AWR Design Environment® platform design wizards are Dynamic Link Library (DLL) files that allow you to automate routine tasks or implement add-on tools to extend AWR Design Environment platform capabilities. The following sections describe the available wizards.

You can author design wizards using Microsoft's flexible ActiveX® technology, which provides a mechanism for creating visual forms for design wizards using any development environment that supports ActiveX, such as Microsoft® Visual Basic®, Microsoft Visual C++®, or Borland® C++Builder®.

To implement a design wizard to be hosted within the AWR Design Environment platform, you create an ActiveX DLL project whose class module implements the IMWOWizard interface. The user interface is implemented by creating a form in the ActiveX control that displays when the IMWOWizard's "Run" method is called.

To run the wizard from within the AWR Design Environment platform, you must register the DLL file as an AWR Design Environment platform add-in. Each registered wizard then displays as a subnode under **Wizards** in the AWR Design Environment platform Project Browser. To run a wizard, double-click its node. When a wizard is run within the AWR Design Environment platform, a wizard state object is created under that wizard subnode. If you enter data into the wizard, you have the option of saving its current state or restoring it to its previous state. To restore the wizard to its previous state, double-click the wizard state object in the Project Browser. To save its current state into the wizard state object, choose **Save State**.

13.1. Amplifier Model Generator Wizard

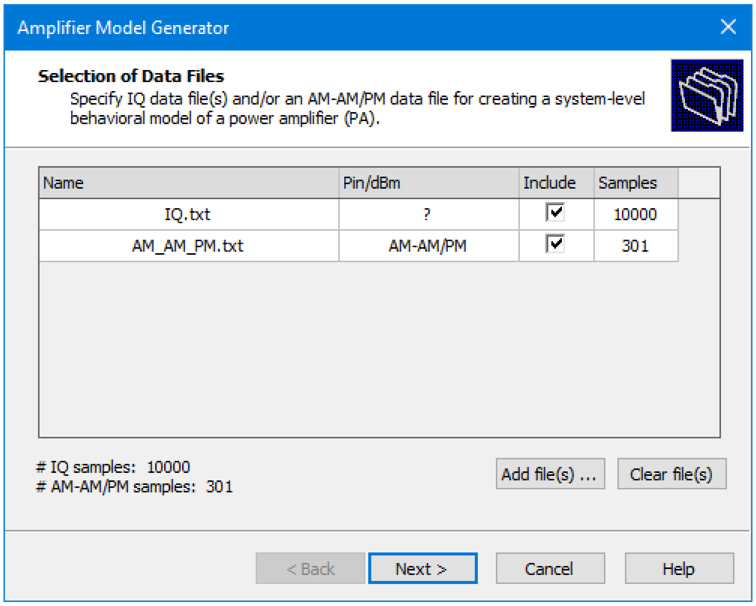
The Amplifier Model Generator Wizard allows you to generate an appropriate system-level nonlinear behavioral model of an amplifier, typically an RF Power Amplifier (PA). This wizard reads in IQ and AM–AM/PM text data files, carries out memory-effect estimation, and generates one of the following two models:

- Frequency Dependent Behavioral Amplifier (File-Based): AMP_F (memoryless)
- Amplifier Model (Time-Delay Neural Network-based): AMP_TDNN (with memory)

To access the Amplifier Model Generator wizard, create a new project or open an existing project, open the **Wizards** node in the Project Browser and double-click **Amplifier Model Generator**.

13.1.1. Selecting Data Files

To add data files, you should have IQ and/or AM-AM/PM text data file(s) generated by measurements or simulations. To select these files, click the **Add files(s)** button and in the Select Data Input File(s) dialog box, navigate to their location, select the file(s) and click **Open**.



By default, the **Include** check box is selected for each added file (and for any optional IQ file input-power level), so all of the files (and power levels) are used for model generation. You can clear this check box to exclude a file from model generation or click the **Clear files(s)** button to clear all of the files.

The simplest way to use the Amplifier Model Generator wizard is to specify one IQ text data file only. This file must conform to the text data file format. The following is an example of a minimalistic IQ file that the wizard accepts:

```
TSTEP = 2.5e-9

 (I,)              (Q,)             (I,)             (Q,)
-0.071310         0.061610         -0.559396        -1.062530
-0.091102         0.079394         -0.699122        -1.334756
...               ...              ...              ...
```

The IQ file must contain the following three sections:

1. The time step tag TSTEP along with its value (or alternatively, the sampling frequency SMPFRQ = 1/TSTEP).
2. The column headings "(I,)" and "(Q,)" (or alternatively, "(Re,)" and "(Im,)") for the in-phase and quadrature components (real and imaginary parts) of the complex input and output signal, respectively.
3. The actual column data (the in-phase and quadrature components of the input and output IQ-signal values).

This wizard requires that this kind of IQ file contains at least 100 IQ input–output samples, although many more (for example, 1000…10000) samples are needed to obtain good modeling results.

Referencing the text data file format, you can extend the IQ file by including comments "! …". You can also provide additional information by using other existing tags. Finally, you can define one or more input-power levels, PIN, followed by the corresponding IQ-data blocks. Here, PIN is the average power of the IQ input waveform of the specific IQ data block. Adding all these, the IQ file might look similar to the following:

```
! IQ data: I_in, Q_in, I_out, Q_out
! # power levels: 2
! # samples per power level: 5000
! # all samples: 10000
```

```
! PIN_start: -5 dBm
! PIN_stop: 0 dBm
! PIN_step: 5 dB

TSTEP = 2.5e-9
CTRFRQ = 1.0e9
SMPSYM = 8
Z0 = 50

PIN = -5 dBm

 (I,)            (Q,)          (I,)          (Q,)
-0.071310        0.061610      -0.559396     -1.062530
-0.091102        0.079394      -0.699122     -1.334756
...              ...           ...           ...

PIN = 0 dBm

(I,)             (Q,)           (I,)          (Q,)
-0.126809        0.109559       -0.994599     -1.880855
-0.162004        0.141184       -1.188688     -2.248207
...              ...            ...            ...
```

Note that the wizard ignores the comment lines "! …". Also note that here, "CTRFRQ = 1.0e9", "SMPSYM = 8", and "Z0 = 50" serve as comments; the tags TSTEP and PIN are the only ones the wizard uses. The tag PIN is used as a separator of IQ-data blocks, and its actual numerical value is only used for printing the value of Pin/dBm in the table. The numerical value of TSTEP, in turn, is written to the possibly generated AMP_TDNN model file.

This wizard requires that each PIN block contains at least 100 IQ input–output samples, however many more samples per each PIN block are needed to obtain good modeling results. You can specify many IQ files, however a more compact, and probably more convenient, approach is to specify just one IQ data file that may (or may not) contain several PIN blocks.

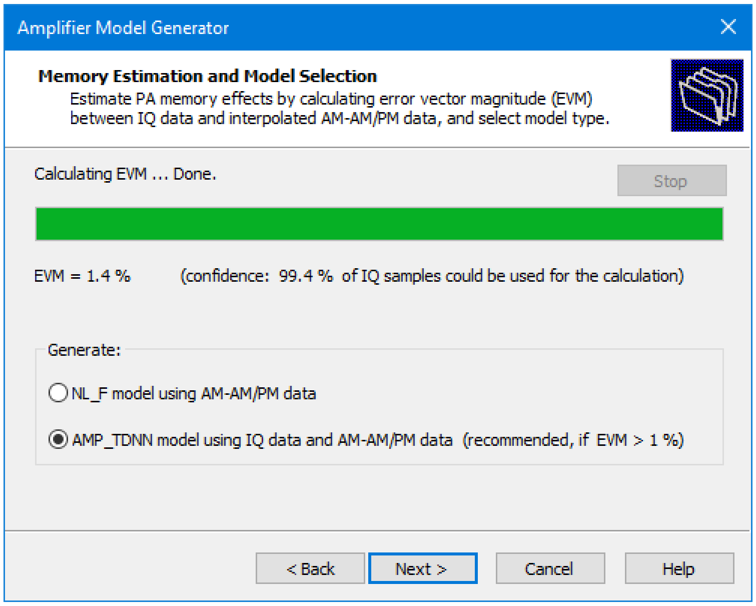
Whether or not you specify IQ data, you can specify one AM–AM/PM text data file, which must obey the text data file format, too. The following is an example of an AM–AM/PM file the wizard accepts:

```
! AM to AM and PM characteristics

Pin(Mag,dBm) Pout(Mag,dBm) PhsVout(Phs,deg)
-20      3.14948      143.601
-19.9      3.24948      143.601
...      ...      ...
```

The options for including text data files and the corresponding actions are:

- If you include only an AM–AM/PM file (and then click the **Next** button), on the following "Memory Estimation and Model Selection" screen, the memory estimation is skipped and the generation of only an AMP_F model is enabled.
- If you only include one or more IQ files (or their power levels), the wizard directly displays the "TDNN Training" screen in order to generate an AMP_TDNN model.
- If you include both an AM–AM/PM file and one or more IQ files, all of the steps on the "Memory Estimation and Model Selection" screen are executed, and you can select either the generation of AMP_F or AMP_TDNN.

13.1.2. Memory Estimation and Model Selection



The Amplifier Model Generator wizard estimates PA memory effects if you specify both IQ data and AM–AM/PM data, and if the input-power ranges of these two data sets overlap. An Error Vector Magnitude (EVM) is calculated between phase-normalized IQ output data and linearly interpolated AM–AM/PM output data that is converted into IQ form. Conceptually, the EVM calculation corresponds to the following three operations:

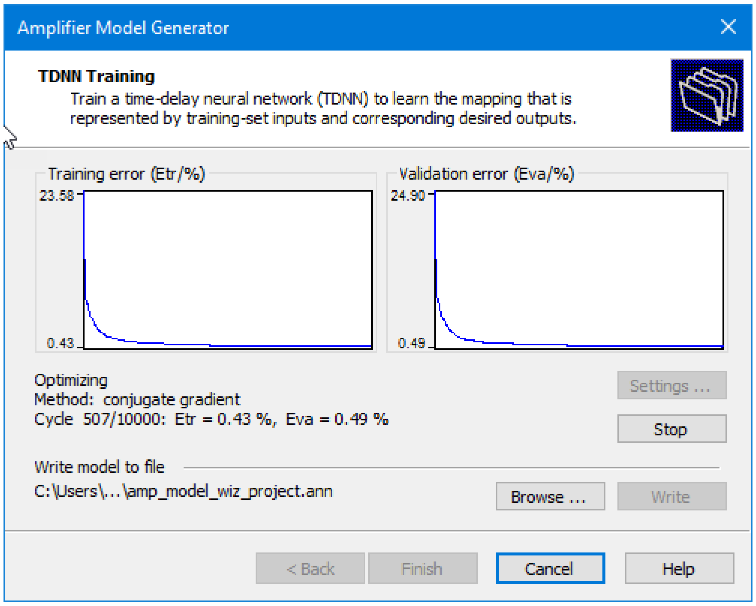
1. The AM–AM/PM data is used to internally create a stripped version of the AMP_F model.
2. For each IQ input, an error vector component between the "AMP_F output" and the desired IQ output is calculated.
3. A normalized EVM is calculated. This EVM tells how well a memoryless "AMP_F model" fits to the IQ data.

The wizard pre-selects an appropriate model as follows: if EVM <= 1 %, the wizard suggests the generation of a memoryless AMP_F model; otherwise, if EVM > 1 %, the wizard suggests the generation of a memory-effect aware AMP_TDNN model. You can change this selection by selecting only the other model.

If you select an AMP_F model and click the **Next** button, the model, which uses the AM–AM/PM data file, is readily launched, and the wizard is closed. (Note that if you have one AM–AM/PM data file, but no IQ data for memory estimation, there is no need to use the wizard; in this case, it is more practical to create the AMP_F model directly.)

If you select an AMP_TDNN model, both the IQ and AM–AM/PM data specified are used for the model generation. Clicking the **Next** button opens the "TDNN Training" screen.

13.1.3. TDNN Training



Prior to starting the TDNN training you should click the **Settings** button to open and review the Settings dialog box to modify the training-related parameter settings if needed.

The next step is to click the **Start** button to start the TDNN training. (Note that this button starts, stops, and continues operations, depending on the status of TDNN training.) You can monitor training progress by viewing the constantly updated training error (Etr/%) and validation error (Eva/%) curves and the following printed information: 1) status, 2) optimization method, and 3) the current optimization cycle vs. maximum number of cycles along with Etr and Eva values.

TDNN training can be stopped for the following reasons:

- Clicking the **Stop** button.
- Both Etr and Eva reach the error goal (see “Settings”).
- The maximum number of optimization cycles is reached (see “Settings”).
- The cross-validation-based early-stopping technique detects, using a safe margin of 1000 cycles, the global minimum of the validation error, Eva. This avoids overlearning (oscillatory over-fitting to (noisy) training data); the TDNN weights corresponding to the global minimum of Eva are stored in memory.

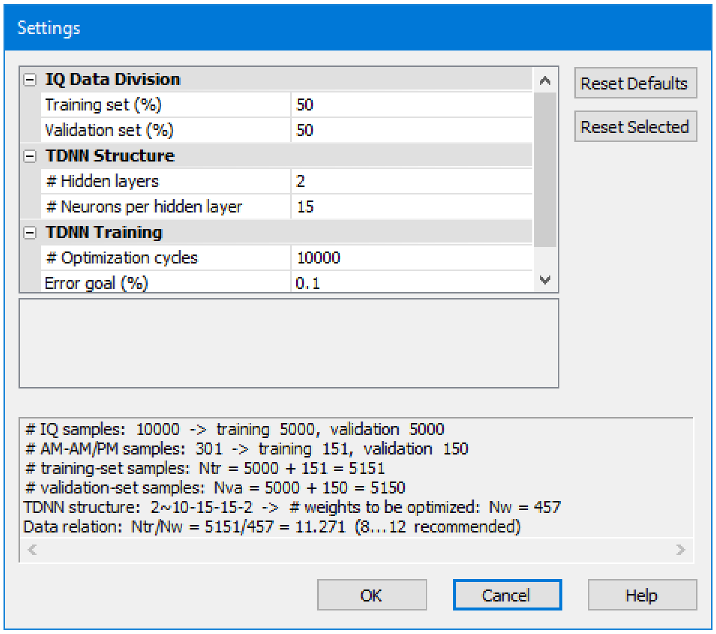
If you stop the TDNN training by clicking the **Stop** button and there are still optimization cycles, you can continue optimization by clicking the **Continue** button.

Generally, if Etr < 1%, then the TDNN fit to the Training data is good. More importantly, if Eva < 1%, then the generalization capability of the TDNN is good.

When TDNN training is stopped, it is possible to write the AMP_TDNN model file. The default location of the model file is the project directory, and the default name of the model file is the project file name with the *.emp* extension replaced by *.ann*. You can change the model file location and name by clicking the **Browse** button. The very first time, the model file is written by clicking the **Write** button. If the file already exists, it is overwritten. The model file includes two parts: 1) prologue comments that partly document the modeling process and 2) inline-comment-like tags along with the actual parameters like weights of the trained TDNN.

After the AMP_TDNN model file is (over)written, clicking the **Finish** button allows the wizard to open the AMP_TDNN model, which uses the desired model file.

13.1.3.1. Settings



In the Settings dialog box, you can review and change the parameter values related to IQ Data Division, TDNN Structure, and TDNN Training. The parameter values are initialized to their default values. To return to these values, click the **Reset Defaults** button. On the bottom of the screen, a text field is updated after each parameter change.

IQ Data Division functions as follows: Training set (%) (default: 50) and Validation set (%) (default: 50) define the percentage of IQ samples picked up into Training and Validation sets, respectively. Specifically, Training set (%) defines, starting from the very first IQ sample of each IQ file or PIN block specified, the percentage of successive IQ samples that go to the TDNN training set. Validation set (%), in turn, defines the percentage of the following successive IQ samples that go to the TDNN Validation set. For example, if you use the default values and have 10000 IQ samples specified, then 50% of them equals 5000 samples, which means that samples 1…5000 go to the Training set, and samples 5001…10000 go to the Validation set. Training set (%) can be specified as 1…100, Validation set (%) 0…99, and their sum is allowed to be 1…100. If Validation set (%) is 0, then no validation is performed; in this case, however, there may be a risk for TDNN overlearning, (oscillatory over-fitting to (noisy) training data).

The division of AM–AM/PM samples is internally derived from the IQ Data Division as follows: If Validation set (%) is 0, (there is no validation data), then all the AM–AM/PM samples go to the Training set; otherwise, if Validation set (%) is 1…99, (there is at least some IQ validation data), then the 1st, 3rd, and so on, AM–AM/PM sample goes to the Training set, and the 2nd, 4th, and so on, AM–AM/PM sample goes to the Validation set. Each AM–AM/PM sample is converted into an equivalent IQ sample before adding it to the Training or Validation set.

Generally, the Training set should contain altogether 1000…10000 IQ samples. If there are less than 1000 IQ samples (per PIN block), then the PA nonlinear dynamics along with short-term and/or long-term memory effects are not captured well enough. Otherwise, if there are more than 10000 samples, then the approximation capability of the TDNN starts to reach its limits.

The AM–AM/PM data is needed for the memory estimation, but its role in the actual TDNN training is less obvious, since the AM–AM/PM data, although properly converted into equivalent IQ data, is rather different from the actual IQ data. Generally, the AM–AM/PM data should be specified if you want the resulting AMP_TDNN model to somehow fit to that data, too. (It is also possible, however, to use the AM–AM/PM data for the memory estimation only, by clicking the **Back** button after the memory estimation and then clearing the **Include** check box for the AM–AM/PM data file.)

You can adjust TDNN structure as follows: **# Hidden layers** (default: 2) selects the number of TDNN hidden layers; 1 or 2 is allowed. **# Neurons per hidden layer** (default: 15) can be 10…30. These two parameters define the TDNN structure, and the number of TDNN weights to be optimized during training. The TDNN structure (here, 2~10-15-15-2), the resulting number of weights (437), and the resulting data relation (11.787) display in the text field. You should verify that the data relation fits into the recommended range 8…12. The notation “2~10-15-15-2” means that the TDNN has two actual inputs (Iin and Qin), 10 intermediate inputs (obtained by signal processing), two hidden layers both having 15 neurons, and two outputs (Iout and Qout).

TDNN Training is the last category. **# Optimization cycles** (default: 10000) is the max number of optimization cycles, which you can set to 1…100000. Generally, at least 1000 cycles are needed to obtain a proper AMP_TDNN model. **Error goal (%)** (default: 0.1), which can be 0.001…10, defines another stopping criterion for TDNN training; if both the training and validation error reach this goal, then optimization is stopped.

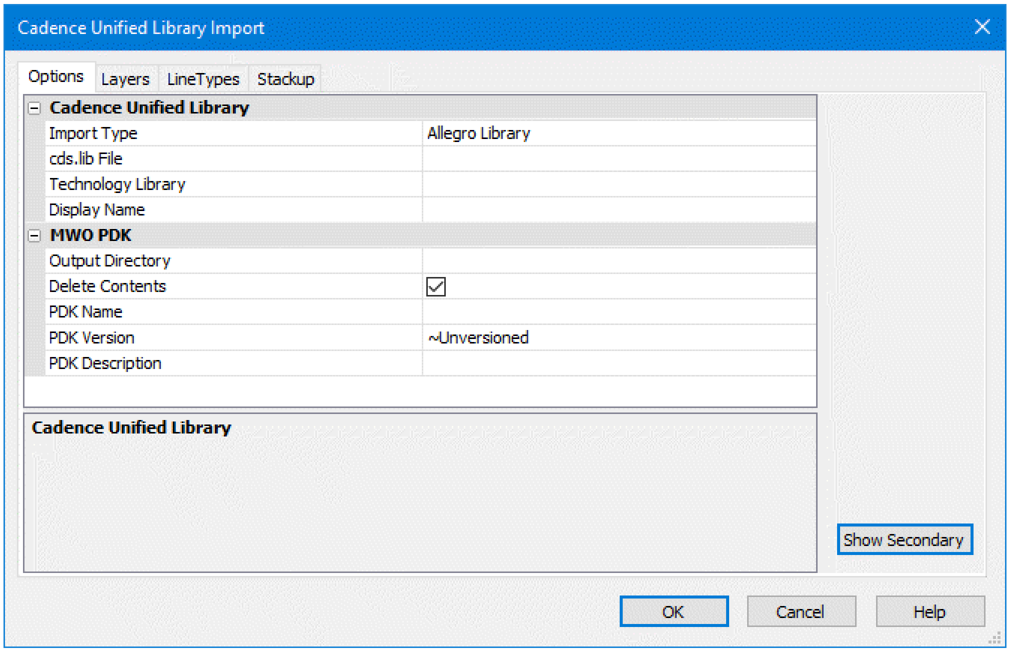
Click **OK** to save the new settings or click **Cancel** to ignore them and return to the previous screen.

13.2. Cadence Unified Library Import Wizard

The Cadence Unified Library Import Wizard allows you to import a Cadence Unified Library from the Allegro or Virtuoso platforms into Microwave Office for creating an RF PCB design. The Unified Library is the foundational block for seamless interoperability between Microwave Office and DE-HDL/Allegro software. This library can contain all of the information necessary to design an RF PCB in Microwave Office software, including the technology on which the RF design is built, and the parts that are available to use. For details about the interoperability between the AWR Design Environment platform and both Allegro and Virtuoso platforms, see Appendix E, *AWR Design Environment Interoperability with Virtuoso and Allegro* . For more information about individual options, click the **Help** button on the dialog box.

To use the Cadence Unified Library Import Wizard:

1. In the Project Browser **Wizards** node, double-click **Cadence Unified Library Import**.

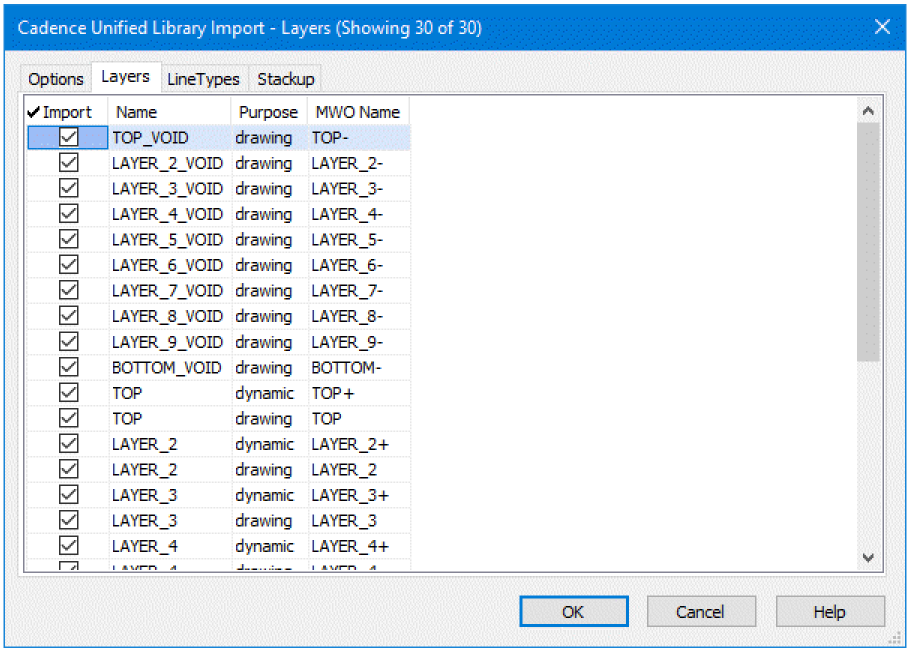


13.2.1. Import Options

The **Options** tab allows you to specify options related to importing a Cadence Unified Library and creating an AWR PDK.

13.2.2. Import Layers Options

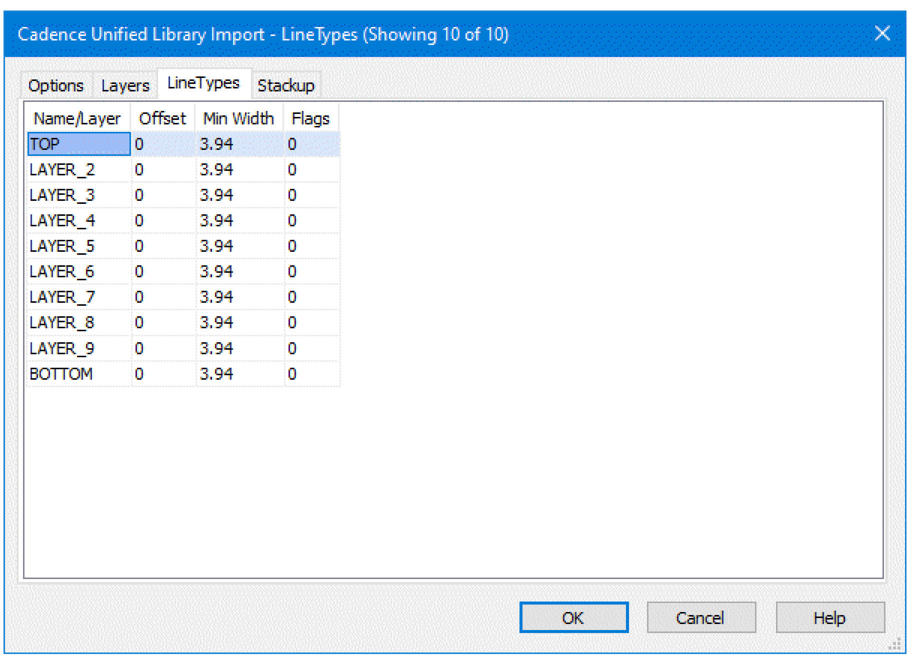
After specifying the import format and related options on the **Options** tab, the import creates a new *.lpf* file containing the layer definitions that are specific to the imported library. You can view the layers on the Cadence Unified Library Import dialog box **Layers** tab.



All columns can be sorted by clicking the column title to toggle between no sorting, ascending, or descending. The grid supports multi-selection using **Shift + Click** for range selection, **Ctrl + Click** for discontinuous selection, and **Ctrl + A** to select all cells in a column.

13.2.3. Import LineTypes Options

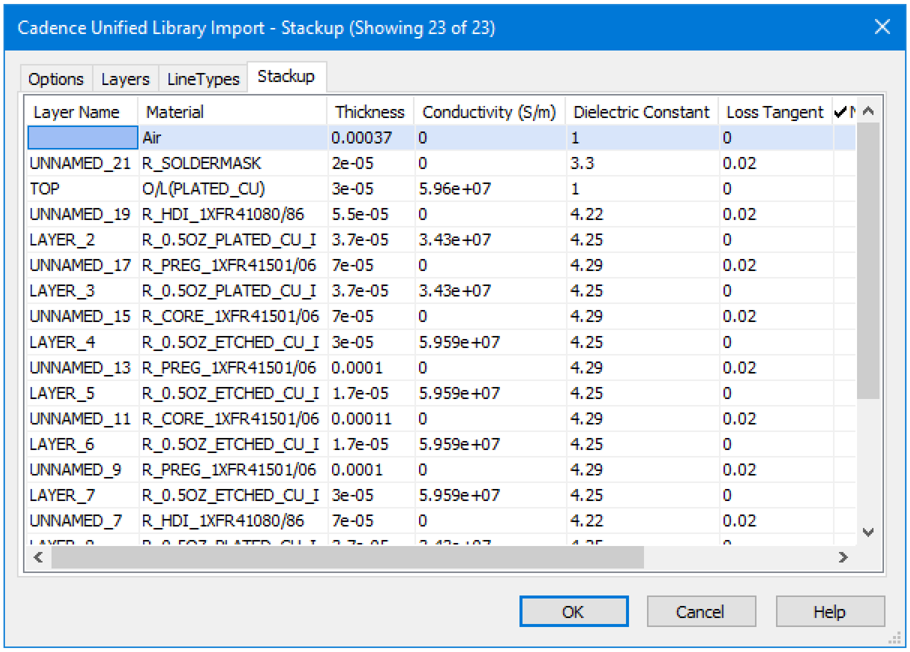
The **LineTypes** tab provides a list of the line types from the LPF file the PDK uses, in the order in which the PDK expects them to appear.



The grid supports multi-selection using **Shift + Click** for range selection, **Ctrl + Click** for discontinuous selection, and **Ctrl + A** to select all cells in a column.

13.2.4. Import Stackup Options

The **Stackup** tab displays the material properties of the stackup found in the design file or synthesized from the file.



13.3. Component Synthesis Wizard

The Component Synthesis Wizard allows you to synthesize several types of passive microwave structures to be implemented in microstrip transmission line structures:

- Multisection **Transformer** - The wizard creates a Chebyshev quarter-wave transformer of one to four sections. The fractional bandwidth can be specified for transformers having more than one section.
- Multisection **Wilkinson** power divider - The wizard creates a Wilkinson divider of one to four sections. The fractional bandwidth can be specified for dividers of more than one section. The input and output impedances need not be identical, but the impedances of the two output ports must be the same and power division must be equal. Fractional bandwidth can be specified for dividers having more than one section. Under **Wilkinson Resistors**, you can select either **Chip** or **Thin Film** and specify the resistivity in ohms/square.

  Resistor values can be calculated accurately only for one- and two-section dividers. For three and four sections, initial values of the third and fourth resistors are estimated from heuristics. Achieving optimum isolation between the output ports requires tuning the resistors. The third and fourth resistors' values are high, sometimes not realizable, and have a relatively weak effect on the isolation. In many cases, the fourth resistor in a four-section hybrid can be simply eliminated.
- **Rat Race Hybrid** - The wizard creates a simple, 180-degree rat race hybrid. You cannot specify bandwidth, but you can enter port resistances, which must be identical.
- Multisection **Branch Line Hybrid** - The wizard creates a 90-degree, branch line hybrid of one to three sections. You cannot specify bandwidth; it is a consequence of the number of sections. All port impedances must be identical. There

exist a number of synthesis methods for multisection branch line hybrids. The type of synthesis used here produces a hybrid having series lines equal to the port impedance and relatively high-impedance shunt lines. As such, it sometimes is not realizable on high-dielectric-constant substrates.

- Multisection Directional **Coupler** - The wizard creates a 90-degree edge-coupled microstrip directional coupler of 1, 3 or 5 sections. Multisection couplers are symmetric. A Chebyshev or Butterworth realization can be specified; fractional bandwidth can be specified only for the Chebyshev design. All ports must have equal impedances. The center section in multisection couplers invariably has high coupling, which in some cases may not be practically realizable. Some experimentation with bandwidth and number of sections may be necessary to achieve a realizable structure.

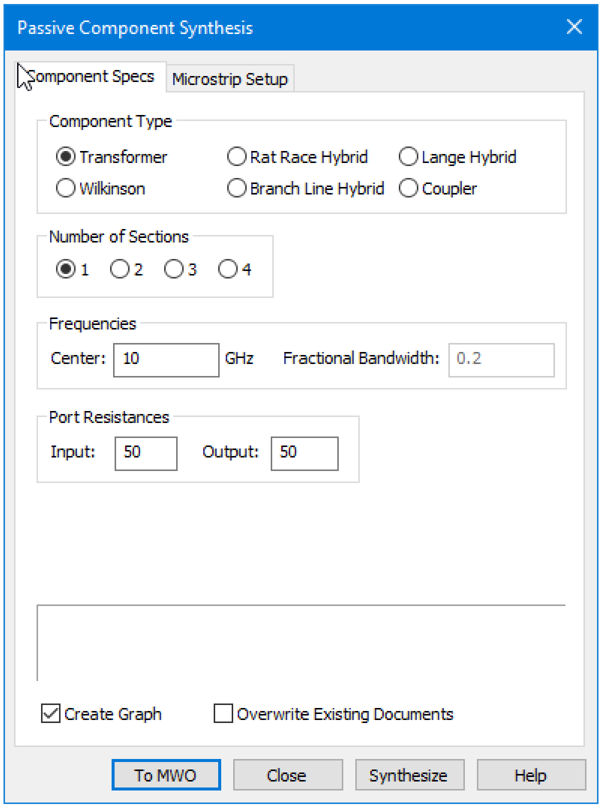
  The coupler design includes bends and straight sections to interconnect the coupled-line sections. Because of the limitations in the bend models, the bend widths and microstrip line widths may not match. Also, at high frequencies, the interconnects may not be short relative to the coupled-line lengths, and they affect the performance significantly. In those cases, it is best to use the synthesis for an initial design and then EM simulation to optimize the coupler. A different substrate may also be necessary to obtain the desired performance.

- **Lange Hybrid** - The wizard creates a 3-dB Lange hybrid, a single-section design. For bandwidth less than approximately 50%, a standard coupler, having even- and odd-mode impedances of 120.7 and 20.7 ohms, respectively, is produced. To achieve wider bandwidth, the hybrid is overcoupled, which results in a coupler having greater bandwidth than the standard coupler, but more coupling variation over that band.

  As with the multisection coupler, the bend and line elements at the ends of the coupled section may be used outside the range of their models’ validity. For this reason, especially at high frequencies, those connectors are best EM simulated. Using a coupled-line model, combined with EM-simulated interconnects, is an efficient and accurate approach to Lange coupler design.

  A wire model is used for the air bridges. Depending on frequency and application it might be best to EM simulate these as well.

To access the Component Synthesis Wizard, open the **Wizards** node in the Project Browser and double-click **Component Synthesis**. The Passive Component Synthesis dialog box displays.



On the **Component Specs** tab, select a **Component Type** and then specify the design parameters:

- **Number of Sections** - Some of the components synthesized by the wizard can have more than one section. For those, specify the number of sections. Components having more sections generally have better bandwidth, but are larger and more complex.
- **Frequencies** - Enter the **Center** frequency, or if relevant, the **Fractional** bandwidth (the bandwidth divided by the center frequency). If the nature of the design is such that you cannot specify bandwidth, this text box is grayed.
- **N Section Coupler** - Depending on the **Component Type**, you can select either a **Chebyshev** or a **Butterworth** design, or specify different input and output impedances.
- **Create Graph** or **Overwrite Existing Documents** - Select an option to create a new graph or overwrite an existing graph.

Click the **Synthesize** button to perform synthesis calculations and display the results in the text window at the bottom of the dialog box. Those data are useful for making sure that the circuit is realizable or for creating a component in a medium other than microstrip. Click the **To MWO** button to synthesize the component and send the design to the Cadence Microwave Office® program.

On the **Microstrip Setup** tab you can configure the microstrip technology parameters for the components:

- **Substrates** - Select a substrate type with its editable default parameters, or select **Global Definitions MSUB**, a substrate that is already defined in the project's Global Definitions and available from the drop-down list.
- Substrate parameters - You can modify the default parameters for the selected **Substrate**.
- **Length Units** - Select the desired length units. It is always best if the units are the same as the project units, as this is the most precise. In any case, the synthesized component's dimension are entered in project units when the Microwave Office project is created.

- **Tee/Step Type** - Specify discontinuity elements that use either the **Closed Form** or electromagnetic (**EM**) models.

The synthesized project includes the Microwave Office schematic, necessary subcircuits, and graphs, but not finished layouts; you need to manipulate the layout to finalize the component design. Appropriate quantities in the subcircuit are entered as tunable variables. Line lengths in the synthesized circuit are modified to compensate for the size of interconnects and bends. That compensation may not be exact in all cases, causing the center frequency to differ from the value entered. To compensate, the synthesized circuit includes a “scale” variable, which proportionately scales the lengths of all relevant microstrip elements, allowing adjustment of the center frequency.

13.4. IFF Import Wizard

The Intermediate File Format (IFF) Import Wizard imports ADS schematics into the AWR Design Environment platform through the use of IFF files.

This wizard displays in the AWR Design Environment platform and runs if you have the proper license feature (IFF_100).

To prepare for export from ADS, when you generate the IFF file for a schematic in ADS you need to enable the **Put a space between numbers and the scalar/unit** option, and choose **Current Design and All Library Parts** for the **Schematic Hierarchy Option**. ADS version 2011.10 or later is recommended.

To access the IFF Import Wizard, open the **Wizards** node in the Project Browser and double-click **IFF Import**. In the IFF Import dialog box, select the *.iff* file to import and specify whether or not to also import a layout IFF file. The imported schematic opens in a schematic window, and an "IFFImport" node displays under the wizard in the Project Browser. Additional imports are named "IFFImport*n*" with sequential numbers. You can change this node name by right-clicking and choosing **Rename**. The AWR Design Environment platform Status Window displays any error messages and a link to the log file. Click on the link to see all information, warning, and error messages.

13.4.1. Import Options

There are three options in the dialog box that affect the import process:

- **Include file portion of circuit names in schematic names**: In the schematic IFF file the workspace/project file name is incorporated into the names for the schematics. Select this option to strip out the file names from the names when creating the schematics.
- **Do not use pCells during layout import**: Layout IFF files normally contain both pCell instances and the base primitives for drawing the pCells. Select this option to draw the layout using only the primitives.
- **Create single-layer Line Types for all layout layers**: The layers on which a pCell draws are often determined by selecting a Line Type for the pCell. If the project into which the IFF files are being imported does not have appropriate Line Types already defined, select this option to automatically create Line Types for each layer defined in the layout IFF file.

13.4.2. Advanced Import Options

Click the **Advanced** button in the IFF Import dialog box to display additional import options:

- **Wiring Options**:
  - **Connect unattached wire ends to nearest component node** - Wires are snapped onto a grid in AWR Design Environment platform schematics, which can cause wires not to attach to a component, and therefore not be removed during the normal schematic clean-up process. Select this option to automatically stretch unattached wire segments to the nearest component node.
  - **Draw arrows to modified wire segments** - Draw arrows to the locations where wire ends were moved on the schematic.

- **Draw arrows to locations of omitted wire segments in schematics** - If a wire segment is too short or diagonal it cannot be drawn in an AWR Design Environment platform schematic. A warning message is issued. Select this option to draw an arrow to the location of an omitted segment on the schematic.

- **Library Options**:
  - **Import for library creation** - Enable features specifically for component library creation. For example, when only a layout IFF file is imported, schematics and layouts with duplicate names are ignored.
  - **Create list of symbol origin offsets (SymbolOffsets.dat)** - Creates a text file (named *SymbolOffsets.dat*) that lists the coordinate offsets of the first pins of the imported schematic symbols.

- **Miscellaneous Options**:
  - **Scale schematic coordinates by factor of** - Scales the coordinates in the imported schematic by the value specified. This can be helpful for getting components and wires to land on grid.

13.4.3. Component Mapping

There are several ways to map ADS components into Microwave Office components. This wizard includes code for mapping many models. You can use mapping files to extend the mapping to additional models or override the built-in mapping code. These files may be in three different locations:

- In the installation directory */Wizards* folder an *IffImport_element_map.txt* file contains any "factory" mappings that are not hard-coded.
- In a PDK, the *.ini* file can point to a mapping file using a line similar to the following:

```
[File Locations]
IffImportMap=Library\ADS_element_map.txt
```

- In the AppDataUser directory there can be another *IffImport_element_map.txt* file, for user-defined mappings.

These files are read in the order listed. If there are mappings for a particular component in more than one of the files, the mapping read last takes precedence. The individual map entries in these text files allow you to specify the name of the ADS model and the name of the Microwave Office model into which it should be mapped. You can also map parameter names and alter the terminal order. The syntax for the entries is documented at the top of the *IffImport_element_map.txt* file in the */Wizards* directory. Alternatively, an entry can map an ADS model to a SUBCKT element that references either a schematic defined in the project or an AWR Design Environment platform *.sch* (exported schematic) file. (See the *IffImport_element_map.txt* file for details.) If an IFF file contains a component type that is not mapped by the mapping files or the built-in mapping code, the wizard tries to map it to an Microwave Office model with the same name as the ADS model. If no such model exists, the wizard creates a template schematic for the component in the project and places an associated SUBCKT element in the schematic being imported. You can then fill in the details in the template schematic to manually map the component.

The wizard has built-in mapping code for the following ADS models:

- AC, AgilentHBT_NPN, AgilentHBT_NPN_Th, Angelov_FET
- Balun4Port, BJT_NPN, BJT_PNP, BJT4_NPN, BJT4_PNP, BONDW1, BONDW2, …, used with BONDW_Usershape, BSIM4_NMOS, BSIM4_PMOS
- C, CAPQ, CLIN, CLINP, CCCS, CCVS
- DAC (referencing DSCR file), DC, DC_BJT, DC_FET, DC_Block, DC_Feed, Diode
- EE_HEMT1
- GaAsFET (Curtice Cubic, Advanced Curtice Quadratic, Modified Materka, and Statz)

- HarmonicBalance, HICUM_NPN, HICUM_PNP, Hybrid90, Hybrid180,
- I_Noise, V_Noise, I_NoiseBD, I_Probe, INDQ, I_DC
- L
- MACLIN, MCROSO, MCURVE, MCURVE2, MSABND_MDS, MSOBND_MDS, MBEND, MGAP, MLEF, MLOC, MLIN, MOSFET_NMOS, MOSFET_PMOS (BSIM3 only), MSTEP, MSUB, MTAPER, MTEE, MTEE_ADS, MUC2, MUC3, ..., MUC10, Mutual
- P_1Tone, P_nTone, ParamSweep, PLCQ, PRL, PRC, PLC, PRLC,
- R, RCLIN
- S_Param, S1P, S2P, ..., S99P, S1P_Eqn, S2P_Eqn, S4P_Eqn, SCLIN, SCROS, SCURVE, SBEND, SDD1P, SDD2P, ..., SDD9P (with I[n,0], I[n,1], or F[n,0] only), Short, SIMKIT_MM_PSP_NMOS, SIMKIT_MM_PSP_PMOS, SIMKIT_MM_PSPe_NMOS, SIMKIT_MM_PSPe_PMOS, SLCQ, SLIN, SLEF, SLOC, SLSC, SMITER, SRL, SRC, SLC, SRLC, SSTEP, SSUB, STEE, SweepPlan
- Term, TermG, TF, TLIN4, TLIND, TLIND4, TLINP, TLINP4, TLOC, TLSC, TLPOC, TLPSC
- V_1Tone, V_AC, VAR, VBIC_NPN, VBIC_PNP, VBIC5_NPN, VBIC5_PNP, VCCS, VCVS, V_DC, VIA, VIAFC
- XFERP, XFERTAP

13.5. IFF Export Wizard

The Intermediate File Format (IFF) Export Wizard is used for sending schematics and layouts to Cadence PCB tools, Keysight/Agilent, Mentor, or Zuken software using IFF files.

This wizard displays in the AWR Design Environment platform and runs if you have the proper license feature (IFF_100).

To access the IFF Export Wizard, open the **Wizards** node in the Project Browser and double-click **IFF Export**. In the IFF Export dialog box, select the schematic to export and specify whether or not to also export the associated layout IFF file to the designated path.

13.5.1. Export Options

There are two options in the dialog box that affect the export process:

- **Export sub-schematics**: Include in the export any associated sub-schematics.
- **Include element definitions**: Include in the export all element definitions.

13.5.2. Advanced Export Options

Click the **Advanced** button in the IFF Export dialog box to display additional export options:

- **Omit ARTCOMP commands from layout IFF** - Do not include ARTCOMP commands in the layout IFF file.
- **Mentor RFMsg DLL** - Displays the file name for the RFMsg DLL used for the Mentor “RF Connect” design transfer tool.

13.6. iFilter Filter Wizard

The Cadence® iFilter™ filter synthesis wizard displays in the Cadence AWR Design Environment® platform if you have the proper license file (FIL-200, FIL-250, FIL-300, or FIL-350) to run the wizard.

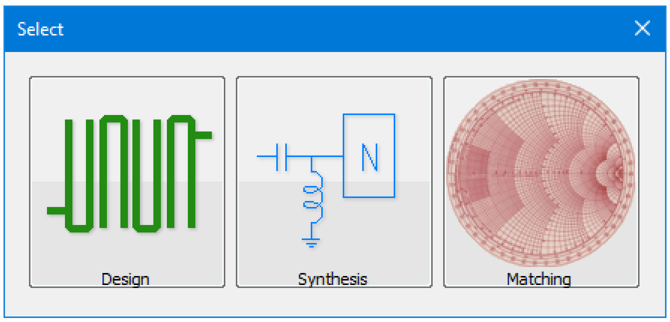
13.6.1. Using the iFilter Wizard

The iFilter Wizard uses the same interface for all filter types, the main iFilter dialog box. From this dialog box you can access all options and settings.

13.6.1.1. Starting the iFilter Wizard

You can run the iFilter Wizard to create a new filter or to modify an existing filter.

To create a new filter, open the **Wizards** node in the Project Browser and double-click **iFilter Filter Synthesis**.



- To run Standard iFilter, click **Design**.
- To run Advanced iFilter with synthesis capabilities, click **Synthesis**.
- To run the Cadence iMatch™ impedance matching wizard for designing impedance matching networks, click **Matching**.

You can also display this Select dialog box from within the program by clicking the **Select Design Mode** button. 

To modify an existing filter, right-click the filter under the **iFilter Filter Synthesis** node in the Project Browser and choose **Edit**.

The main iFilter dialog box displays with properties from any previous design.

13.6.1.2. Running the iFilter Wizard

While the wizard displays, you can change filter and approximation type, edit specification and technology parameters, and configure other options. After every change, the wizard recalculates the element values, redoes the realization (layout or part selection) and calculates and plots the response. You do not need to press a special button after modifications as all the views are kept current.

To enter a different parameter value or specification in the main iFilter dialog box you can use the keyboard, click the up/down arrows next to an option to increase/decrease values, or use the mouse wheel (click in the desired edit box and scroll the mouse to increase/decrease the value). The step size is automatic based on the type and value of the edit box. Press **Ctrl** while scrolling to increment/decrement with a smaller step size.

13.6.1.3. Closing the Wizard

To close the iFilter wizard:

- Click **Generate Design** to create a schematic, graph(s) and other items in the AWR Design Environment platform. A filter design item displays under the **iFilter Filter Synthesis** node in the Project Browser.

- Click **OK** to create a filter design item under the **iFilter Filter Synthesis** node in the Project Browser only. No schematic, graph(s) or other items are created.
- Click **Cancel** to close the main iFilter dialog box without saving.

13.6.1.4. Design Properties

In various stages of the wizard, new designs are created, previous designs are recalled or existing designs are modified. To preserve continuity, the wizard continually transfers data between changes. If you start a new design, the latest filter design properties are loaded into the new design rather than prompting you to re-enter all specifications. Also, when you change a filter type, all of the common specifications such as passband corners and technology settings are copied to the new design.

13.6.2. Filter Design Basics

This section provides a brief introduction to filter design with a focus on iFilter terminology. It is not intended to cover every aspect of filters. For more information about designing electrical filters please see a complete reference source.

13.6.2.1. Approximating Function

An ideal filter that passes a desired frequency range with no loss, and stops all undesired frequencies with no leakage is impossible to obtain. It is therefore necessary to approximate a filter response by using some filtering functions (transfer functions) that yield realizable element values, like Chebyshev or Maximally Flat filters. Some of these functions provide sharp stopband attenuation, while others provide flat group delay in the passband. Selecting an approximating function is always a trade-off, while most of the time Chebyshev meets most of the desired characteristics of filtering.

Filters represent varying input impedances with respect to frequency. Depending on the phase and magnitude of the impedance, they either pass or reject frequencies. A series capacitor, for example, has infinite impedance at f=0 Hz which causes all the signals to reflect at zero frequency. A series inductor, on the other hand, has infinite impedance at f=infinity which does not pass any signal.

Transmission Zero (TZ)

A transmission zero is defined as the frequency where no signal transmission occurs. For cascaded element filters (ladder type), TZ's can be created by infinite impedance series elements or zero impedance shunt elements. For example, a series inductor or a shunt capacitor can create a TZ at f = infinity, letting these elements be used for lowpass filters.

Finite Transmission Zero (FTZ)

A finite transmission zero is defined as TZ where frequency is a finite number as in 0 <f< infinity. FTZ's are created by infinite impedance series elements or zero impedance shunt elements at a finite frequency. For lumped element lowpass filters, a series-LC-resonator (SLC) connected as shunt to the circuit creates an FTZ. Likewise, a parallel-LC-resonator (PLC) in the shunt arm also creates an FTZ. You can use both of these elements for lumped-element lowpass or bandpass filters. Not all of the filter types allow FTZ's, however, because the realization technique used may not be suitable.

Monotonic Filters

A filter with no FTZ is sometimes called a "monotonic" filter. For a given filter order, monotonic filters provide less selectivity than filters with FTZ(s). However, their ultimate stopband (regions far away from the passband) contain less spurii.

In iFilter, available approximations for monotonic filters are:

- Chebyshev

- Maximally Flat
- Bessel
- Linear Phase
- Gaussian
- Transitional Gaussian
- Legendre

Chebyshev filters provide maximum stopband attenuation for a given filter degree. As a consequence, there is a ripple in the passband of the filter. As the ripple gets smaller, the relative stopband attenuation reduces (the filter becomes less selective). Maximally Flat filters do not have a passband ripple but have a good selectivity. The other approximations are mainly used for pulse shaping rather than providing selectivity. Bessel filters, for example, provide an excellent flat delay within the passband. All of these approximations are based on the performance of lowpass prototypes. Once these basic filters have been transformed into the other passbands, such as bandpass, they tend to lose their attractive properties.

Unless the selected filter type does not pose any other limitations, the generic filter degrees are available up to 50 for Maximally Flat and Chebyshev filters. For very high order filters, the element values start losing accuracy due to computation limitations. For other approximations, the maximum filter degree is 10.

Filters with FTZ(s)

In iFilter, the available approximations with FTZ's are:

- Elliptic
- Generalized Chebyshev

Elliptic filters have ripples both in passband and stopband. Elliptic filters can provide very sharp selectivity at the expense of wide-spread element values. Because they are not monotonic, they also have a finite attenuation at high frequencies.

Generalized Chebyshev filters are a good balance between monotonic and elliptic filters. Their FTZ's are all concentrated at one single frequency and they tend to result in better element value spread than elliptic filters. There are two available types:

- Type 1: all TZ's are one single frequency with 1 TZ at infinity
- Type 3: all TZ's are one single frequency with 3 TZ's at infinity

13.6.2.2. Filter Synthesis

Filter synthesis is the process of constructing an approximating transfer function, then a driving point impedance function, and finally extracting elements from the impedance function. Transfer functions involve poles and zeros in normalized S-domain. For most designers, poles and zeros of the transfer function do not mean much. In reality, cascaded filters with no cross-coupling have explicit pole-zero locations that can be found in textbooks. iFilter, therefore, uses the TZ/FTZ terminology which correspond to the jw-axis, a measurable frequency quantity.

For monotonic and well-behaved FTZ functions, prototype tables and explicit formulas exist. iFilter uses this "LP-Prototype" approach to generate filters. A more generalized filter design includes transmission zero placement and element extraction. This method is called “filter synthesis” and it’s covered in Advanced iFilter product.

13.6.2.3. Design using LP-prototypes

There are two major lowpass prototypes: Ideal LP and Microwave. Ideal LP-prototypes are shown as series inductors and shunt capacitors. Microwave prototypes also include impedance/admittance inverters and they are mainly used for narrowband bandpass filters.

Prototype element values are also called "G-values". G-values are calculated through explicit formulation, or obtained using references such as Zverev's Handbook of Filter Synthesis. Values are normalized to a source resistance of 1.

G-values are then converted into selected passband of the filter. This process is called "Frequency Transformation". An inductor prototype element is frequency-transformed into:

- an inductor for lowpass filters
- a capacitor for highpass filters
- a series LC-resonator for bandpass filters
- a parallel LC-resonator for bandstop filters

Likewise, a capacitor prototype element is transformed into:

- a capacitor for lowpass filters
- an inductor for highpass filters
- a parallel LC-resonator for bandpass filters
- a series LC-resonator for bandstop filters

Inverters are replaced by inductive or capacitive Pi- or Tee-sections for bandpass filters.

The last step of the filter design is Impedance Renormalization. Inductive parts are multiplied, and capacitive parts are divided by the actual source resistance, which is usually 50 ohms.

Although this design method is given for lumped element filters, distributed element filters can be derived in a similar way as described in the following section.

13.6.2.4. Distributed Element Filters

Distributed elements filters are obtained by using cascaded transmission lines, cascaded coupled lines, or multiple coupled lines. When terminated by a load impedance ZL, a transmission line with a characteristic impedance of Zo has the following input impedance:

$$Z_{in} = Z_o \frac{Z_L + jZ_o \tan\theta}{Z_o + jZ_L \tan\theta}$$

where θ is the electrical length of the line. For transmission lines built as printed circuits, θ can be expressed as

$$\theta = \frac{2\pi f \sqrt{\varepsilon_r}}{c} l$$

Stubs

If the termination of the line is a short circuit (ZL=0), then the input impedance is described by

$$Z_{in} = jZ_o \tan\theta$$

Thus a shorted line shares the same input impedance characteristics as an inductor.

Similarly; an open ended transmission line (ZL=infinity) has an input impedance characteristic as a capacitor,

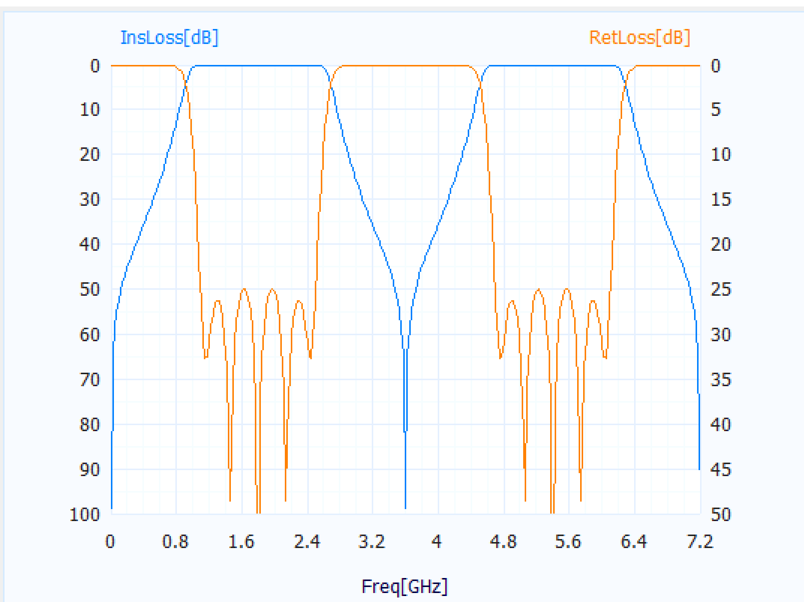
$$Z_{in} = \frac{Z_o}{j\tan\theta}$$

These two special-terminated transmission lines are also called "stubs".

Periodicity

As found in most textbooks, `s=jtanθ` is called Richard’s variable. Tanθ is a periodic function and it repeats itself every θ. As a result, distributed element filters obtained by transmission lines periodically repeat their frequency response.

In the following figure, a distributed highpass filter response is given. The filter is designed with 50deg electrical length at 1 GHz. The elements are 90deg (θ/2) at 90/50*1GHz = 1.8GHz. The filter repeats itself at every 180 degrees (θ), (every 3.6GHz). As shown in the figure, the response from 0 to 3.6GHz is identical to the response from 3.6 to 7.2GHz.



In general, distributed lowpass filters and distributed bandstop filters have the same attenuation characteristics. Likewise, distributed highpass and distributed bandpass filters share the same response.

Filter Design

You can obtain distributed element filters using different methods.

The first method is to design lumped element filters using LP-prototypes and replace inductors and capacitors with stub equivalents. The drawback to this method is the size and practicality. A highpass filter calls for a series capacitor, which is equivalent to an open-circuited transmission line in the series arm. This is not possible in microstrip medium, so the technique is limited to lowpass and bandstop filters only.

The second method is to replace lumped elements in the first method with high and low impedance transmission lines. An inductor can be approximated by a high impedance transmission line in a limited-frequency band. Likewise, a capacitor can be approximated by a low impedance transmission line. This method is used in stepped impedance lowpass filters, but the approximation is only as good as the extreme impedances that can be obtained.

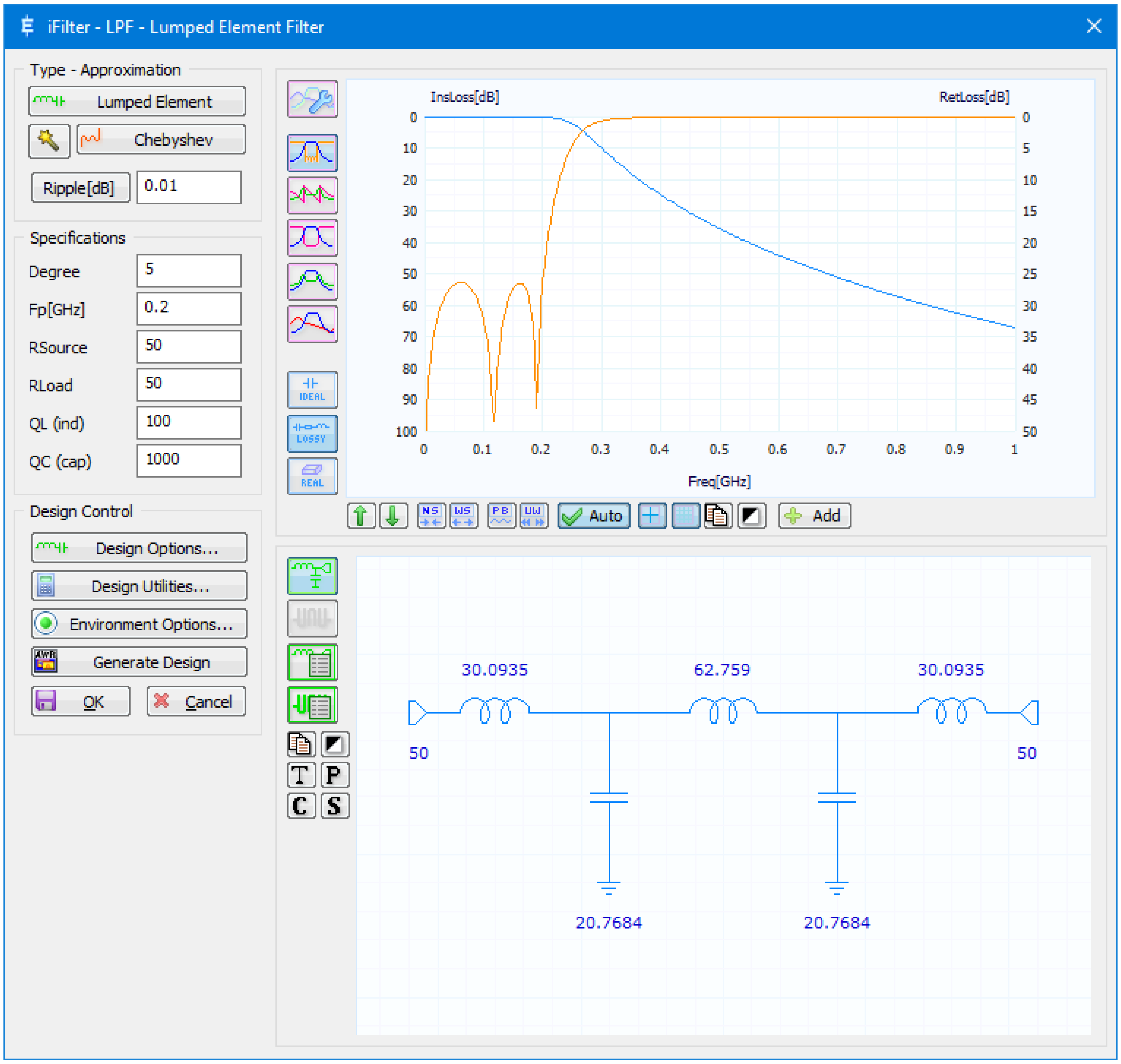
The last method is to use coupled transmission lines, either by direct coupled synthesis or by using the coupling coefficient method. There are various types of microwave resonators, such as end coupled, edge coupled, and interdigital. In this method, resonators are either synthesized or selected and coupled using gaps between them. The amount of coupling is dictated by synthesis formulas. Calculating gaps for given coupling is cumbersome, and it involves explicit formulations as well as lengthy EM simulations. However, the end result is rewarding, as this method yields very compact circuits with excellent selectivity.

13.6.3. General Flow of Filter Design

This section describes how to use the iFilter Wizard in the usual filter design process flow.

13.6.3.1. Main iFilter Dialog Box

The main iFilter dialog box is the control center of the iFilter Wizard.

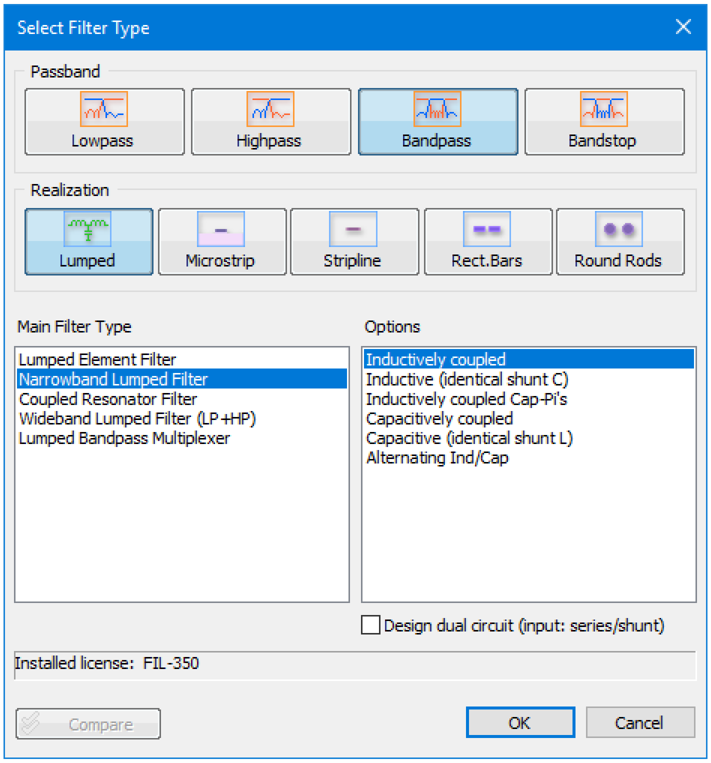


This dialog box is resizable and registers its last state during runtime, so it opens with the size and location at which it previously closed.

The top left-hand side of the dialog box allows you to select the filter type, approximation, filter order and other electrical specifications. The bottom left-hand side of the dialog box allows you to change the physical parameters of the filter, save, and close the wizard. The top right-hand side shows the response of the filter and the associated chart control. The bottom right-hand side provides a view of the equivalent circuit schematic and physical layout. All items in the dialog box, including the layout and the plot, are current with the specifications. You do not need to click a button to re-design or re-analyze.

13.6.3.2. Select Filter Type Dialog Box

To exit the current design and create a new filter in the main iFilter dialog box, click the **Filter Type** button. The button is at the top left, labeled with the currently selected filter type.



Select **Passband**, **Realization** and **Main Filter Type** in order. Most of the filters have optional types that present subtle variations.

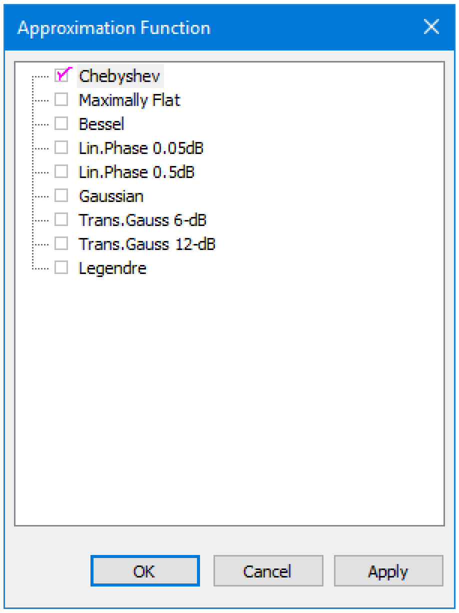
For some filter types, you can design the filter with a series or shunt element at the source side. In this case, a **Design Dual Circuit (input: series/shunt)** check box is available. If you select this option for a lowpass filter, for example, the design starts with a shunt capacitor rather than a series inductor. A shunt capacitor, however, is only available when source resistance is greater than or equal to load resistance, (for example, RS >= RL). For series inductors, RS <= RL is needed. If the source and load resistances are specified otherwise, the only possible option is selected regardless of the status of this check box.

Click **OK** to change or confirm the filter type. Properties of the existing filter are copied to the new filter.

NOTE: Some technologies may not be available due to license limitations. Contact Cadence for license arrangements.

13.6.3.3. Approximation Function Dialog Box

To select the filter approximation function (transfer function), click the second button from the top left in the main iFilter dialog box. The button is labeled with the currently selected approximation function.



Almost all filters can have monotonic approximations, but only some filter types can contain finite transmission zeros. Only available approximations for the selected filter type are listed.

For monotonic filters, there are limits beyond which numeric errors build up or the response cannot be made any more selective. The available filter degree range is based on the following, unless the selected filter topology dictates otherwise.

- Maximally flat, Chebyshev: 1 to 50
- Elliptic types: 3 to 15
- Generalized Chebyshev: 5 to 15 (odd-number). Even orders are not possible with passive elements.
- For all other types, such as Bessel and Gaussian, the order is limited to a range of 1 to 10. Although these approximations are rarely used, they are included in iFilter for completeness.

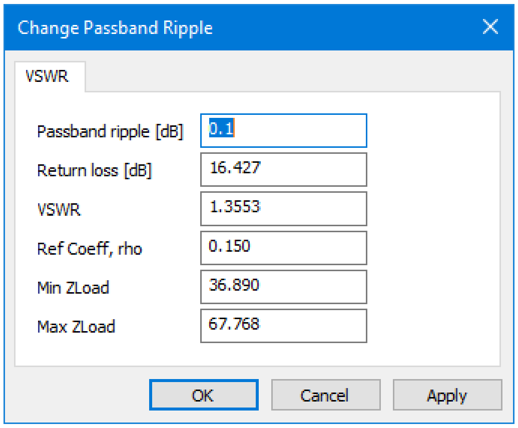
In most cases, the available filter order is overridden by the selected filter topology. For distributed filters, first order filter is almost redundant so it is simply skipped. Likewise, a 50-resonator edge coupled filter is not feasible, so the maximum order is set to 15 for that type.

13.6.3.4. Change Passband Ripple Dialog Box

The Chebyshev approximation is used in the majority of filter designs because the Chebyshev approximation supplies reasonable element values and provides significant selectivity among the monotonic filter group. Elliptic and Generalized Chebyshev types provide more selectivity, however not all of the topologies are suitable for finite TZ's and the response is more sensitive to element variations.

For Chebyshev type, you can specify the passband ripple. Passband ripple is a trade-off between matching and selectivity. If passband ripple is decreased, the passband return loss increases (not desired) and stopband selectivity increases (desired).

You can type the passband ripple in the text box, or click the **Ripple[dB]** button to display the following dialog box.



Passband ripple [dB], **Return loss [dB]**, and **VSWR** are all related parameters. You can enter any of them and the wizard calculates and uses the corresponding passband ripple in dB. In this dialog box, the values for **Min Zload** and **Max ZLoad** are informational only; they display the impedance termination in order to create the VSWR shown when used in a 50-ohm system.

13.6.3.5. Modifying Specifications

When you select the filter type and approximation type, the relevant parameter editing boxes display in the main iFilter dialog box. You can click an edit box, type, and press **Enter** to validate the entry, or you can click another parameter instead of pressing **Enter**.

After you change a parameter value the filter is automatically redesigned. The filter response is plotted and the schematic and layout are updated.

Common specification types are:

- **Degree** - This is the prototype degree or number of resonators for a filter.
- **Fp** - Passband corner for Lowpass/Highpass filters (S21=Ripple dB for Chebyshev, Elliptic and Generalized Chebyshev, and S21=3.011dB for all other approximations).
- **Fo** - Passband center for Bandpass filters, Stopband center for Bandstop filters.
- **BW** - Passband width for Bandpass filters, Stopband width for Bandstop filters. This is measured from the ripple or 3-dB corner as previously explained.
- **Stopband IL** - Peaks in the stopband for elliptic type approximation. Specifying a high value provides very high attenuation, but the selectivity is not very sharp. A low value may provide sharper attenuation near the passband corner, however it may result in unrealistic element values and the ultimate stopband peaks are very high.
- **Lshunt** - Shunt inductor value for lumped, capacitively coupled resonator bandpass filter.
- **Low Zo**, **High Zo** - Lowest and highest allowed impedances for distributed type lowpass filters.
- **Reson Zo**, **Line Zo** - Internal impedance levels for microwave filters.
- **Lshunt** - Shunt inductor value for lumped, capacitively coupled resonator bandpass filter.
- **RSource** - Source termination (left-hand side of schematic)
- **RLoad** - Load termination (right-hand side of schematic)
- **QL**, **QC**, **TLatt** - Parasitic and loss factors for elements for simple analysis (see "Distributed Model Options Dialog Box" for more information.)

13.6.3.6. Analyzing a Design

Every time you change a filter specification it is concurrently re-designed and analyzed, and its response is plotted. The iFilter Wizard offers three types of analysis:

- Ideal
- Lossy
- Real

You select Analysis mode by clicking the following buttons, located to the left of the plot.

Analysis Mode	Lumped Element Filters	Distributed Element Filters
IDEAL	Elements are analyzed as IND, CAP	Elements are analyzed as lossless TLIN and shorted/open circuited stubs.
LOSSY	If SRF is disabled, elements are analyzed as INDQ, CAPQ models. If SRF is enabled, elements are analyzed as INDQP, CAPQP models.	Not available
REAL	Elements are analyzed using the models selected on the Lumped Model Options dialog box, **Realization** tab (click the **Design Options** button). Models include real vendor data.	Elements are analyzed as lossy TLN.

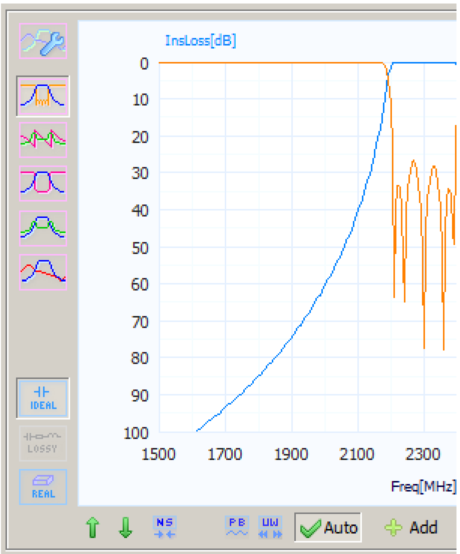
Analysis mode also determines how the design is generated when you click the **Generate Design** button.

Analysis Mode	Lumped Element Filters	Distributed Element Filters
IDEAL	Elements are mapped to IND, CAP	Elements are mapped to lossless TLIN, and shorted/open circuited stubs.
LOSSY	If SRF is disabled, elements are mapped to INDQ, CAPQ models. If SRF is enabled, elements are mapped to INDQP, CAPQP models.	Not available
REAL	Elements are mapped using the models selected on the Lumped Model Options dialog box, **Realization** tab (click the **Design Options** button). If a model does not include SRF, INDQ/CAPQ mapping is used. If model includes SRF, INDQP/CAPQP mapping is used. Real vendor data is also mapped according to this criteria.	For multi-coupled line circuits like Interdigital, Hairpin, and MxCLIN/SxCLIN, mapping is used. Otherwise, MLIN/SLIN/MCLIN/SCLIN type cascaded transmission lines and coupled lines are used in mapping.

NOTE: Cadence does not accept liability for the accuracy of third-party party models, as such data is only available from vendors. However, Cadence is dedicated to providing the best design software. Cadence communicates often with component vendors and progressively updates Cadence simulation models.

13.6.3.7. Plotting Response and Chart Control

On the right-hand side, the filter response is plotted. Analysis is specifically tailored to filter design; therefore only popular measurements are available.



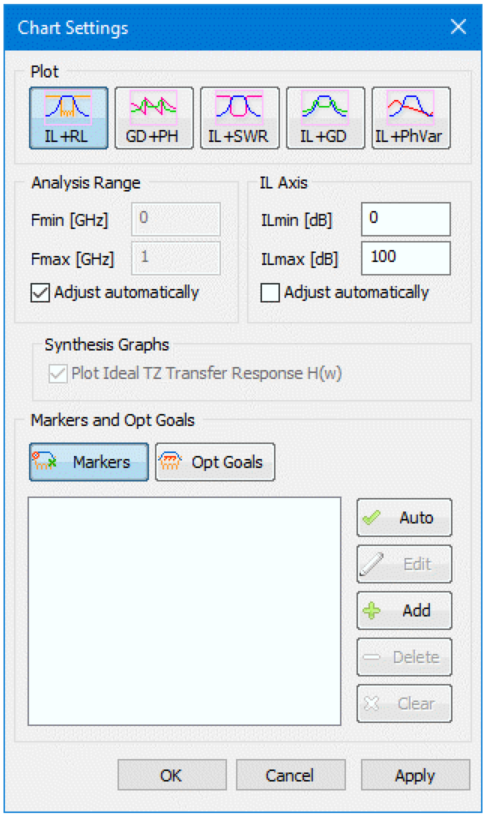
The five buttons below the **Plot Settings** button set the chart, auto-scale the axes, and add the associated measurements to the plot. The axes are always auto-scaled for simplicity. The iFilter Wizard analyzes circuits in a split second, therefore changing frequency span or type of measurements is not a deterrent to the designer. After analysis, the left Y-axis is scaled to a reasonable range and the right Y-axis is scaled up accordingly. The X-axis range is usually rounded to reasonable values, however, the internal /div scale may not always fit to a reasonable grid count.

The small buttons below the plot setting change the frequency span (X-axis) of the chart. Depending on the passband type (for example, lowpass/bandpass) and bandwidth and corners, span is calculated automatically for narrow/wide/ultrawide buttons. The **Passband Analysis Span** button scales the X-axis to filter passband only, to see the loss profile of the passband. Clicking the **Auto Span** button automatically changes the span when **Fp**, **Fo** or **BW** is changed.

The **Add Marker** button adds markers to the chart. You can also add markers by clicking the **Edit Chart Settings** button at the top of the main iFilter dialog box and then clicking the **Markers** button in the Chart Settings dialog box.

13.6.3.8. Chart Settings Dialog Box

To access the Chart Settings dialog box, click on the top button to the left of the filter response.



The first row of buttons select a preset response combination. In filter design, the following response definitions are used more often than standard S-parameters:

- Insertion Loss (IL) = -dB|S21|. IL > 0 for passive filters.
- Return Loss (RL) = -dB|S11|. RL > 0 for passive filters.
- Voltage standing wave ratio (VSWR) = (1+|Rho|) / (1-|Rho|) where Rho is the reflection coefficient.
- Insertion Phase (PH) = Ang(S21)
- Phase Variation (PhVar) = variation of phase around a hypothetical linear phase.

The following preset chart types are available for plotting:

- Insertion Loss and Return Loss
- Group Delay and Insertion Phase
- Insertion Loss and Input VSWR
- Insertion Loss and Group Delay
- Insertion Loss and Phase Variation

In the **Analysis Range** section, you enter the minimum and maximum frequency range of the analysis. If the **Adjust range automatically when frequency specs are changed** check box is selected, iFilter automatically sets the analysis range when you change Fo, BW or Fp specs. If you do not want the analysis range to change, clear this check box.

The **IL Axis** section provides controls for the Y-axis scaling for plotting the Insertion Loss. In previous iFilter versions, the IL axis scaling was automatic depending on the extent of data within the frequency range. This behavior is still available by selecting the **Adjust Automatically** check box. If the check box is cleared, you can enter the Y-axis range and this axis scaling remains fixed while the specifications and frequencies change.

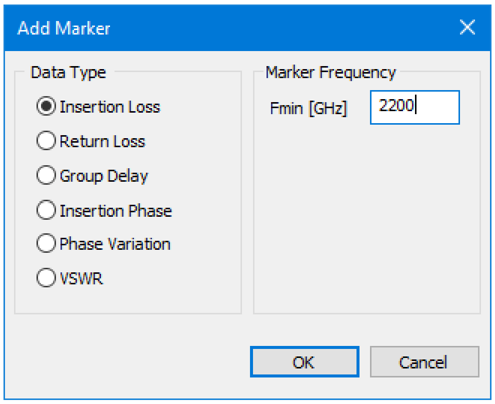
In the **Markers and Opt Goals** section, you can add markers and optimization goals to the filter to visualize the filter response. To add a marker, select **Markers** and then click the **Add** button. To edit or delete an existing marker, select the marker and click the **Edit** or **Delete** button. To delete all markers, click the **Clear** button.

To add, edit, delete, or clear optimization goals, select the **Opt Goals** button and perform the same action.

To speed up the design process (recommended), you can click the **Auto** button to provide most of the common values for the selected filter type. For example, markers are added to the passband center and corners of a bandpass filter. For the same bandpass filter, optimization goals are added for minimum insertion loss in the passband, and 50dB attenuation is added in the upper and lower stopband.

13.6.3.9. Add/Edit Marker Dialog Box

To add or edit a marker, click the corresponding button in the Chart Settings dialog box.



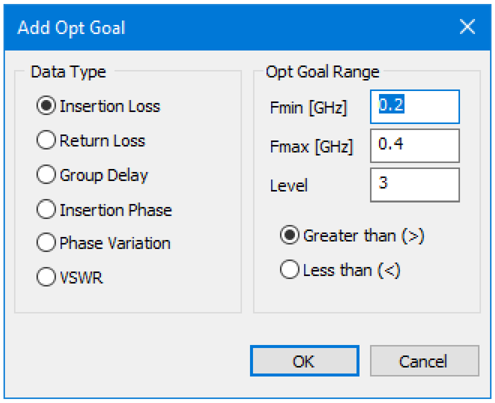
Select a **Data Type** and enter a **Marker Frequency** for the desired marker. You can add markers for any data (measurement) type, but they only display when corresponding data is plotted.

NOTE: In the main iFilter dialog box, you can move markers without opening the Add/Edit Marker dialog box. In the chart area, scroll the mouse-wheel up and down to change the marker frequency. If there is more than one marker, right-click until the desired marker is marked with an "X".

13.6.3.10. Add/Edit Opt Goal Dialog Box

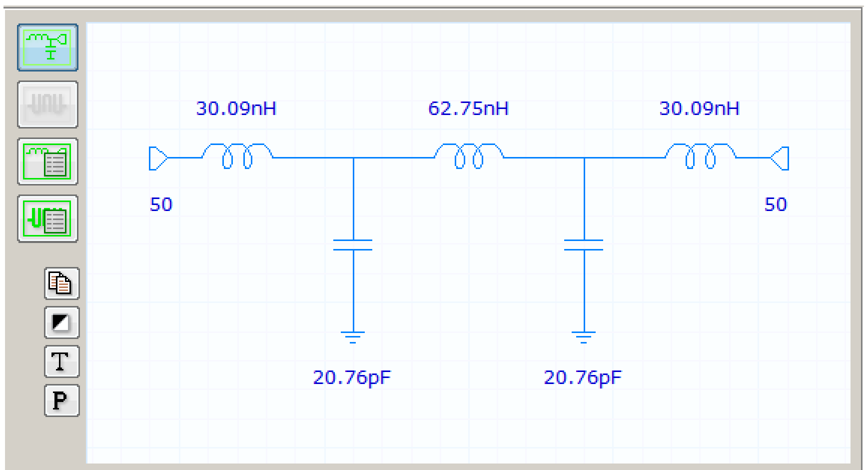
In the Add/Edit Opt Goal dialog box, **Fmin** and **Fmax** are the goal range and the **Level** is the criteria. **Greater than** and **Less than** determine which side of the criteria is desired. If you create an Opt Goal for **Insertion Loss**, and **Insertion Loss** is a positive number (for example, IL = -S21dB), in the passband, IL should be smaller than a maximum loss level so you should select **Less than**. For stopband, IL should be greater than a required attenuation so you should select**Greater than**.

You also use Optimization Goals in setting up Optimization blocks when you generate the design in the AWR Design Environment platform.



13.6.3.11. Viewing the Schematic and Layout

On the bottom right-hand side of the main iFilter dialog box, schematic and layout information display depending on the selected button.



The top two buttons (**View Circuit Schematic** and **View Layout**) display the schematic or layout. The schematic is for circuit elements, either in lumped or distributed transmission line form. For lumped element filters, the schematic is the actual filter and there is no associated layout. For distributed element filters, the schematic represents an equivalent circuit from which you can calculate a response. A physical layout displays how the filter will appear, but most of the time you should perform an EM analysis on physical layouts. To increase speed, iFilter only calculates schematic-based responses. Some loss and parasitic information can be included in the analysis. For lumped element filters, the response is exact. For distributed element filters, the schematic-based response is exact for transmission line types and reasonably accurate for coupled line types.

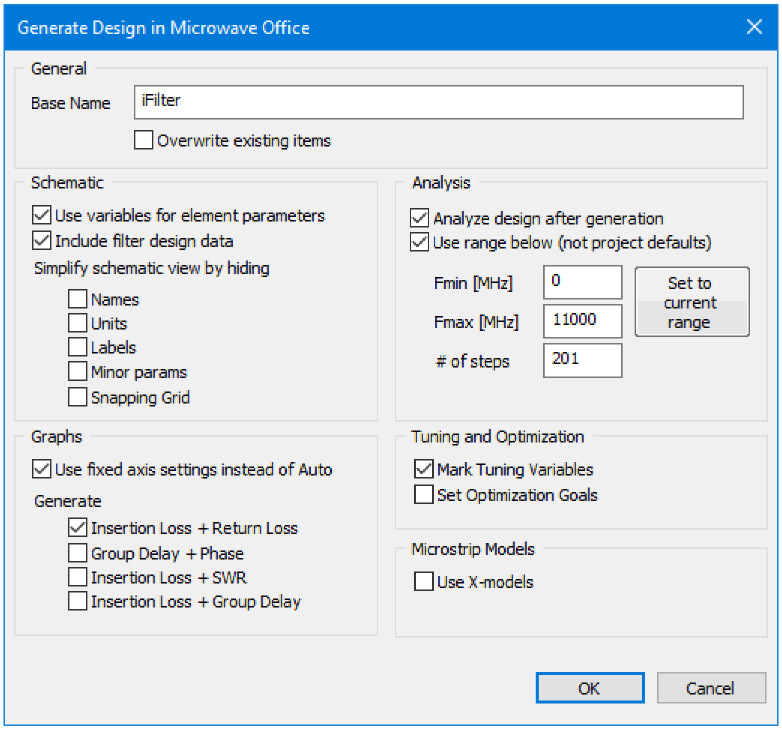
If a warning displays, such as when a parameter value is close to a limit, the corresponding layout element displays in orange. If a limit violation occurs, the element and the **View Layout** and layout info (**View Physical Dimensions**) button display in red. Schematics rarely have warnings. In normal conditions, all buttons display in green.

The four small buttons next to the schematic area are for copying information into the Clipboard, toggling the schematic/layout colors between a light or dark background, turning on/off the text display, and toggling the Properties dialog box display. The Properties window shows the details for the selected element in the schematic/layout.

13.6.3.12. Generate Design Dialog Box

To use the extensive analysis capabilities of the Cadence AWR Design Environment software such as statistics, yield, and optimization, you should generate a design in terms of Cadence Microwave Office® software components: schematic, layout, and graphs. This is exporting an iFilter Wizard item to the Project Browser.

To generate a design in the AWR Design Environment platform, click the **Generate Design** button near the bottom left of the main iFilter dialog box to display the following dialog box.



General Section

Under **General**, type the **Base Name** of generated items such as schematics and graphs. You can also use an existing name, although a warning displays to tell you that the exported item already exists. To overwrite the existing item and turn off the warning, select the **Overwrite existing items** check box.

Schematic Section

In the **Schematic** section, you set the exported schematic options. You can use variables for parameters. When a parameter is generated as a variable, it is defined as an equation on top of the schematic, and the parameter is referenced to that equation. This is particularly useful when there are common parameters of symmetric elements in the circuit. To generate equations in the schematic and assign parameters to them, select the **Use variables for element parameters** check box.

You can hide some schematic element properties such as **Names**, **Units**, and **Labels** to simplify the view. If you select the **Minor params** check box, non-essential parameters in the schematic are hidden. You can also hide the schematic viewing grid by selecting the **Snapping grid**check box.

NOTE: The iFilter Wizard generates a schematic based on the current analysis settings. For more information on analysis settings, see "Analyzing a Design".

Analysis Section

A generated schematic typically requires an analysis. To analyze the schematic after generating it, select the **Analyze design after generation** check box.

The AWR Design Environment platform can analyze all design items with the same frequency range settings (project defaults), or analyze them individually by setting them at the schematic level. You can use the default settings by clearing the **Use range below (not project defaults)** check box, or set your own range by selecting this check box and typing the values. When you open this dialog box, the frequency range from the current analysis range displays. To copy this range, click the **Set to current range** button.

Graphs

The Microwave Office program has an extensive list of measurements and graph types. For designing filters, you only need a limited subset. iFilter allows you to plot various responses and generate them in the Microwave Office program. By selecting chart types (graph types), you can preset graphs and associated measurements.

You can leave the AWR Design Environment platform to scale the Y-axes depending on the range of the associated measurements, or by selecting the **Use fixed axis settings instead of Auto** check box, you can allow iFilter to set the Y-axes.

Tuning and Optimization

iFilter can export tuning variables and optimization goals derived from the current design.

If you select the **Mark Tuning Variables** check box, iFilter determines the major circuit element parameters and sets them as tuning variables. If you tune the exported circuit in the Microwave Office program (press **F9**), the tuning variables are set during export.

If you select the **Set Optimization Goals** check box, iFilter exports the Opt Goals as optimization goals in the Microwave Office program. To add or edit Opt Goals, click the **Edit Chart Settings** button in the main iFilter dialog box to display the Chart Settings dialog box, and then click the **Opt Goals** button. iFilter also sets the optimization parameters in the Microwave Office program which are the same as the tuning variables. iFilter defines a rough constraint (20% above and below) for bounding the values.

Microstrip Models

iFilter exports Standard model and X-model microstrip elements. Select the **Use X-models** check box if you prefer X-models. You can select the following models using this option:

Standard: MTEE$, MSTEPO$, MBEND2$, MLEF, MOPEN, MCROSS$

X-model: MTEEX$, MSTEPX, MBEND90X$, MLEFX, MOPENX, MCROSSX$

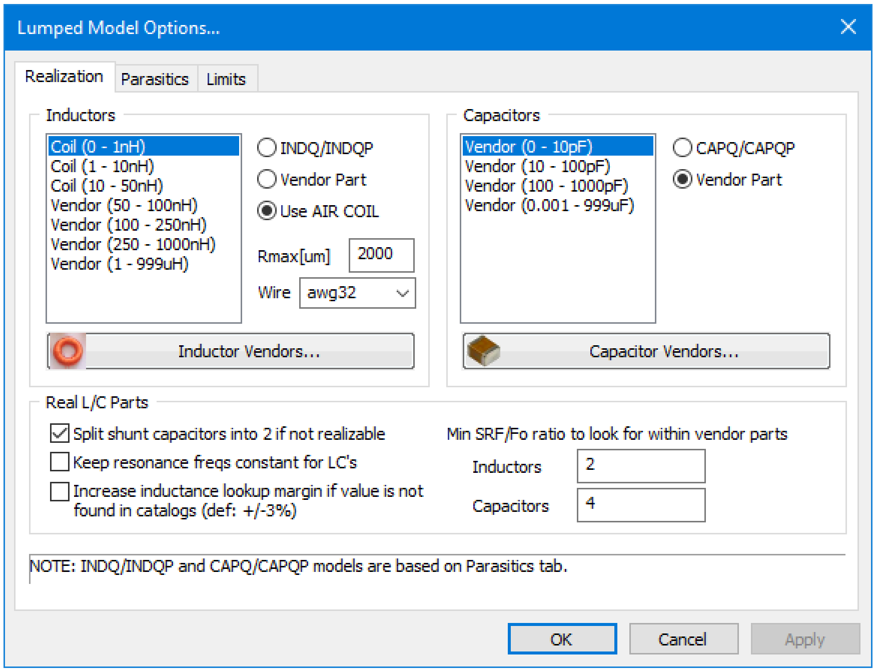
After selecting all relevant options, click **OK** to generate the selected items in the Microwave Office program and close the main iFilter dialog box.

13.6.4. Lumped Model Options Dialog Box

Lumped element filters are initially designed with ideal inductors and capacitors. To build a lumped element filter, you should substitute real life components for ideal elements. In iFilter, you select real life components in the Lumped Model Options dialog box. To open the Lumped Model Options dialog box, click the **Design Options** button while designing a lumped element filter.

13.6.4.1. Lumped Model Options Realization Tab

You can edit inductor, capacitor, and Real L/C parts options on the Lumped Model Options dialog box **Realization** tab. To view this tab, click the **Design Options** button in the main iFilter dialog box and then click the **Realization** tab.



When iFilter designs a lumped element filter, it holds two sets of circuits:

- Ideal filter with L,C elements
- Real filter with selected models and vendors

You can analyze the ideal filter either lossless or with lossy elements and/or parasitics. In the main iFilter dialog box, three buttons control the analysis type.

The **Analyze Ideal** button is for analyzing the ideal lumped element filter with lossless elements. The output is a textbook type response. The **Analyze Lossy** button is for analyzing the ideal filter with lossy elements and parasitics. See "Lumped Model Options Parasitics Tab" for details. The **Analyze Real** button is for analyzing the real filter with selected models.

For inductor models, various preset and fixed ranges are available. An inductor of the selected range can be modeled as:

- **Use AIR COIL** (an air-wound coil) - where iFilter calculates the required number of turns based on the maximum coil diameter and wire gauge. iFilter tries to fit the calculated inductors to maximum diameter (bigger coils give bigger Q) and within 20 turns. For calculating coils manually, see "Design Utilities Dialog Box".
- **INDQ/INDQP** - where iFilter treats the element as in the ideal filter case, as lossy and/or parasitics. The INDQ element is a simple lossy inductor. The INDQP element is modeled as an INDQ element with a capacitive effect, and so self-resonant. To set loss and/or parasitics for elements, see "Lumped Model Options Parasitics Tab".

- **Vendor Part** - where iFilter searches its internal vendor database, and chooses the part with the best Q.

An ideal capacitor can be modeled as:

- **CAP/CAPQP** - where iFilter treats the element as in the ideal filter case, as lossy and/or parasitics. The CAPQ element is a simple lossy capacitor. The CAPQP element is modeled as a CAPQP element with an inductive effect, so self-resonant. To set loss and/or parasitics for elements, see "Lumped Model Options Parasitics Tab".
- **Vendor Part** - where iFilter searches its internal vendor database, and chooses the part with the best Q.

Every physical lumped element has an associated self-resonant frequency (SRF). An inductor's reactance increases up to a certain frequency based on L = 2xf and suddenly starts decreasing due to stray capacitances between turns. At f=SRF, the inductor becomes purely resistive, and beyond SRF, it has negative reactance like a capacitor. A capacitor, however, exhibits a positive reactance beyond SRF like an inductor.

Cadence does not recommend using elements beyond their SRF, however, you should know the following:

- Capacitors exhibit the highest Q at the lowest frequencies. As frequency gets higher, Q decreases, causing more insertion loss in the passband of filters. In RF and microwave filters, multilayer capacitors are mostly used. Normally, capacitor Q's are much higher than inductors, so even when they decrease it is not a concern unless the passband is very narrow. SRF, however, can be a major problem. SRF is a result of inductive properties of the multilayer capacitors that are caused by interlinking wires. When SRF is considered, the effective capacitance can drop significantly, causing a bandpass filter to have a narrower passband at a lower center frequency. You should choose filtering capacitors to operate as far away from their SRF as possible. Cadence recommends using the smallest possible size capacitors, as they tend to have higher SRF. The trade-off is in the Q, as small-sized capacitors may not have high Q.
- Inductors exhibit an interesting Q-value vs. frequency. They tend to have a Q increasing with frequency up to a point and reduce quite significantly when SRF approaches. Observations show that inductors have their highest Q at about f=SRF/1.5 to 1.7.

In iFilter, you set the SRF criteria on the Lumped Model Options dialog box **Realization** tab. In **Min SRF/Fo ratio to look for within vendor parts**, you can enter a value for iFilter to ignore the undesired practical elements. The value is specified as a ratio of Min SRF/Fo. For example, for a filter designed at 200MHz, a value of 3 directs iFilter to only pick elements from its vendor database that have SRF greater than 3*200 = 600 MHz.

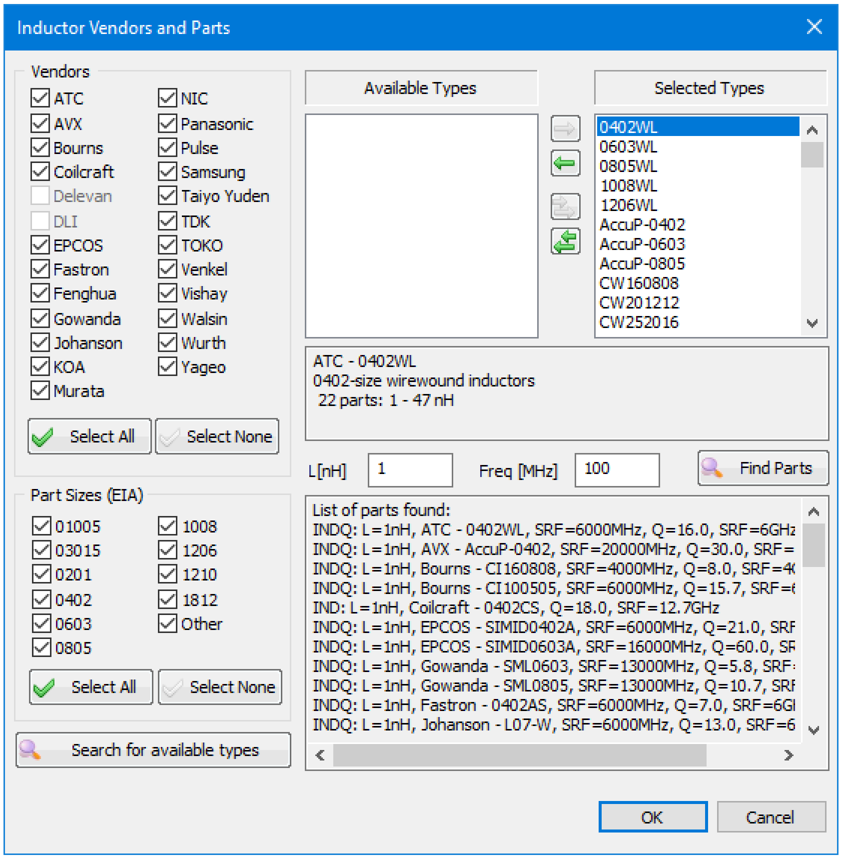
This dialog box also includes the following three options:

- **Split shunt capacitors into 2 if not realizable** - splits capacitors into a maximum of two pieces if the value cannot be obtained with a single vendor part. For example, 5.9pF is not a standard value. If this box option is selected, iFilter searches all of the combinations that can make 5.9pF and chooses the highest Q combination. A combination of 4.7pF and 1.2pF give 5.9pF, for example.
- **Keep resonance freqs constant for LCs** - keeps LC constants the same for each resonator. For LC resonators, iFilter first searches for standard capacitors, as inductors are easier to obtain. If a 2.6pF is needed, and it is only possible to obtain 2.7pF, then iFilter reduces L of the resonator, so that LC multiplication gives the same resonance frequency.
- Increase inductance lookup margin if value is not found in catalogs – extends lookup margin when searching elements. Catalog inductors values do not cover a wide range as capacitors, so finding a catalog inductor may sometimes be difficult with the default margin (%3). If this option is selected, the software slowly increases this margin until a suitable inductor is found in the inductor catalogs.

For capacitors and inductors modeled as vendor parts, you should click the **Inductor Vendors** or **Capacitor Vendors** buttons to set up automatic selection of manufacturer parts as described in the following section.

13.6.4.2. Vendors and Parts Dialog Box

The Inductor Vendors and Parts and Capacitor Vendors and Parts dialog boxes allow you to select vendors and parts from the company inventory. To access these dialog boxes click the **Inductor Vendors** or **Capacitor Vendors** buttons on the **Realization** tab of the Lumped Model Options dialog box.



This dialog box contains many options, yet it is simple to use. To select available vendors and parts:

1. Between the **Available Types** and **Selected Types** list boxes, click the double left arrow button to move all selected types to available types.
2. In the **Vendors** section, click the **Select None** button to de-select all vendor parts.
3. Select all vendors that are available.
4. In the **Part Sizes** section, click the **Select None** button to de-select all part sizes.
5. Select all part sizes that are available. Note that the sizes are listed in EIA type which is mostly used in the USA.
6. Click the **Search for Available Types** button.
7. In the **Available Types** list box, select the parts to use for designs, and then click the single right arrow button to move the type to the **Selected Types** list box. You can select all types by clicking the double right arrow button. Some models do not provide an acceptable filter response. You should avoid these parts, as they may be selected for use simply based on their high Q.

NOTE: The iFilter Wizard uses models "as is" (using the manufacturer datasheets). Where individual SPICE models are not provided, iFilter uses simple models based on SRF and Q data. It is not the intention or responsibility of Cadence to match vendor part datasheets with their actual performance in a circuit.

8. The bottom right-hand side of this dialog box is for testing values only. When you enter an element value and frequency and click the **Find Parts** button, iFilter displays all of the found values and lists their associated Q's and SRF properties.

13.6.4.3. Vendor Part Libraries

Lumped element component manufacturers publish catalog data to support their products. There is no standard format for these data sets. For an inductor, for example, Coilcraft publishes their own models (CCIND in Microwave Office), TDK publishes simple equivalent models; and AVX provides part selection software. Some manufacturers also provide S-parameter data sets. Not all the available data is self consistent and often the information provided illustrates the different modeling approaches employed by manufacturers. Some parts catalogs are more generous in providing measured Q values, the equivalent electrical model (in SPICE format) and/or S-parameter data. To a novice filter designer, this can be more confusing than having no data at all, as often these three pieces of information conflict!

Lack of adequate information can be overcome by traditional design techniques. After the part family type is selected, or selected from the company warehouse stocks or available lab kits, a designer can go through charts by interpolation and produce their own data or use past experience with the vendor from previously designed filters and other designs.

Searching for vendor parts amongst various data sheets in a reliable fashion is a challenging task for a filter designer to do manually. The goal of a successful search algorithm built into a filter design tool is to scan a single library of pre-processed parts data in a convenient and controlled manner.

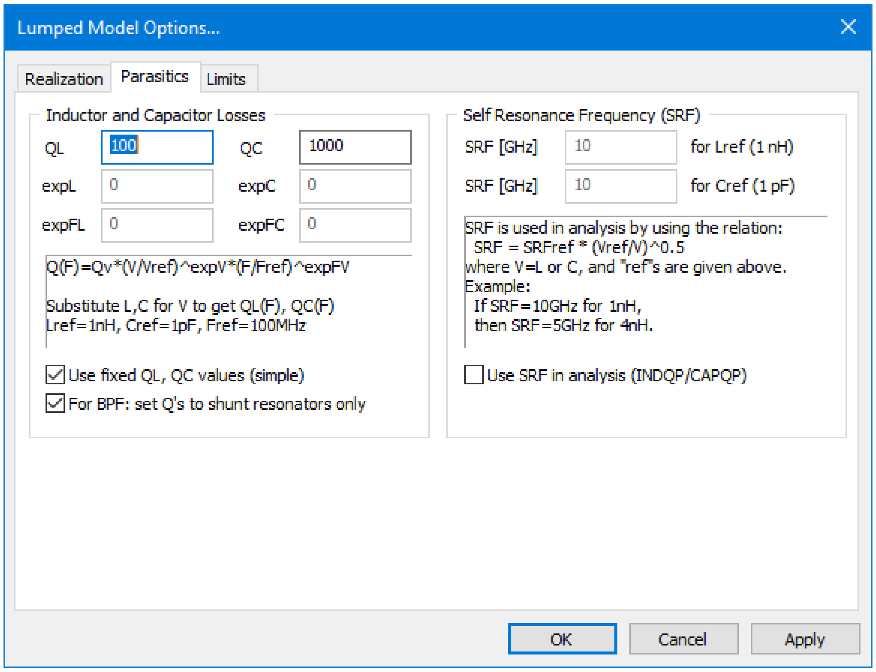
iFilter includes built-in vendor part libraries. The libraries are stored and programmed to allow:

- Nearest model: The available data is interpreted for each vendor and product type separately and they are mapped to the most appropriate component model. For inductor case, it may be a CCIND, an INDQ, and INDQ with self-resonance effects.
- Fast part-search: iFilter is capable of searching its library of over 7000 parts instantly. Every time you change a filter specification, iFilter automatically searches through more than 25 vendors and chooses the part with the highest unloaded Q at the application frequency. On fast computers, the hourglass cursor that displays during filter design may not be visible.
- Automated testing: Because of its optimized library, it takes less than one minute to test the full library. This test assures the integrity and accuracy of AWR software products.

13.6.4.4. Lumped Model Options Parasitics Tab

Practical inductors and capacitors have two major parasitics: loss and self-resonance. Loss is simulated by using series or parallel resistors that are calculated from the element's unloaded Q data. Self-resonance, however, is a limitation of the element and it is simulated by adding a resonating counterpart at the specified frequency.

You can specify inductor and capacitor parasitics on the Lumped Model Options dialog box **Parasitics** tab. To view this tab, click the **Design Options** button in the main iFilter dialog box and then click the **Parasitics** tab.



Losses

In iFilter, Q can be either simple or advanced depending on the option setting. If the **Use fixed QL, QC values (simple)** check box is selected, the top QL and QC values in the dialog box are taken as constant throughout the analysis range. This is a simple but effective approximation of loss.

If the **Use fixed QL, QC values (simple)** check box is cleared, advanced Q settings are assumed.

$$Q_L(f) = Q_{L,base}\left(\frac{L}{L_{REF}}\right)^{\exp L}\left(\frac{f}{f_{REF}}\right)^{\exp F}$$

In the advanced Q settings, Q is a function of frequency and element value. The inductor case is given previously, and capacitor case is intuitive. In the equation:

- base value - is as specified in the **Inductor and Capacitor Losses** section **QL** or **QC** option.
- Lref, Cref - are reference values as 1nH and 1pF.
- Fref - is reference frequency: 100 MHz
- F - is analysis frequency
- L, C - are values of the element in the circuit

For example, assume QL=100 for 1nH and QL=150 for 10nH are given at 100 MHz within available inductor stock. By setting QL=100, expV=0.1761 and expF=0, you can simulate Q’s for any inductor of the given stock. Because,

- QL(f,L) = 100(L=1nH / Lref=1nH)^0.1761 = 100
- QL(f,L) = 100(L=10nH/ Lref=1nH)^0.1761 = 150

If a frequency dependency exists, you can specify expFL to simulate the effect. ExpL, ExpC, ExpFL, ExpFC are all exponents. To specify

- 1/x variation, set -1
- x variation, set 1
- x^2 variation, set 2
- no variation if set as 0 (default)

iFilter also makes QL and QC base values available in the main iFilter dialog box. If advanced Q settings are checked, the main iFilter dialog box changes QL,base and QC,base. If the **Use fixed QL, QC values (simple)** check box is selected, they are used as constant QL,QC values for the simulation.

The **For BPF: set Qs to shunt resonators only** check box is used to predict losses of narrowband bandpass filters. If a narrowband microwave filter is known to be inductively or capacitively coupled, you can design a lumped narrowband filter of the same degree and passband, and by setting Qs to shunt elements only, you can predict the losses of the microwave filter.

Self-resonance Frequency (SRF)

Every lumped element exhibits a self-resonance frequency where its reactance drops to zero and it becomes purely resistive. For a real life inductor, this is equivalent to a parasitics capacitor connected in series with the inductor element. Likewise, a real life capacitor has an associated inductance (as a result of connecting plates together). Beyond the SRF, an inductor behaves like a capacitor, whereas a capacitor behaves like an inductor.

The SRF affects impedance as well as the unloaded Q of the element. For capacitors, QC decreases as frequency increases and as the SRF is approached. You should therefore use capacitors well below their SRF. For inductors, QL increases with frequency until about SRF/1.5 or 1.7, so it is good practice to select inductors (coils) with the SRF about 1.5-1.7 times Fo.

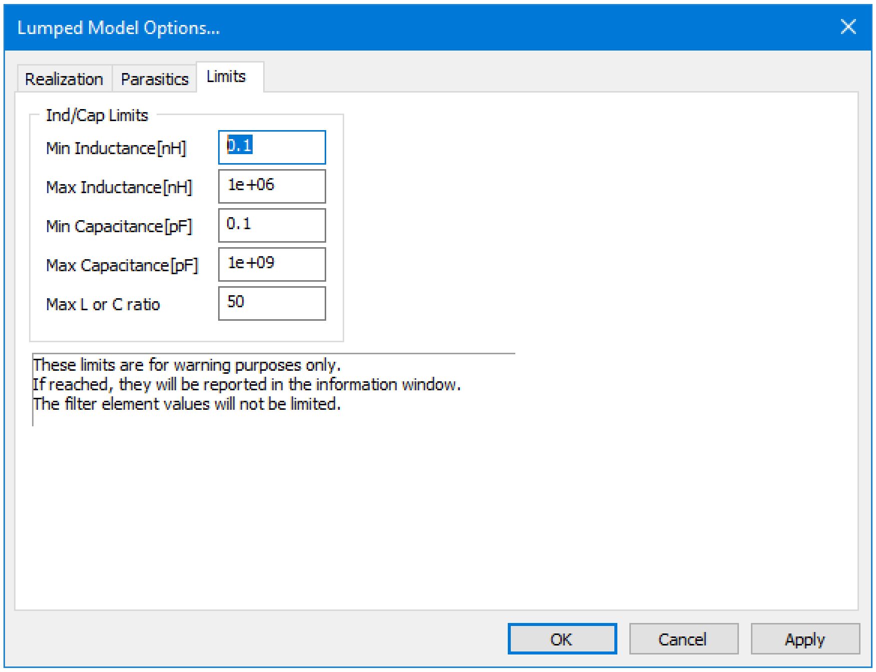
To simulate the effects of the SRF, enter values in the corresponding boxes for reference inductor (1nH) and capacitor (1pF) values. For a given size of elements (for example, 0402, 0805), the SRF tends to change inversely with the square root of the element. For example, if SRF is entered 10GHz for 1nH, it is taken as 5GHz for 4nH. You can use your component vendor's datasheets to find the SRF values for reference values.

When the **Use SRF in analysis (INDQP/CAPQP)** check box is selected, the analysis is performed by adding an internal SRF effect to each element. This is in addition to the losses, if selected. When this check box is selected, and the design is generated in the Microwave Office program, the elements are mapped to INDQP and CAPQP, which have SRF effects. If this check box is cleared, analysis and design generation is based on CAP/CAPQ and IND/INDQ models.

13.6.4.5. Lumped Model Options Limits Tab

You can specify inductance and capacitance warning limits on the Lumped Model Options dialog box **Limits** tab. To view this tab, click the **Design Options** button in the main iFilter dialog box and then click the **Limits** tab.

These limits are used to generate automatic warnings in the View Circuit Information window in the main iFilter dialog box. If the limits are reached, iFilter adds a small warning icon to the tree entry.



For lumped element filters, you can set values for:

- Minimum inductance
- Maximum inductance
- Minimum capacitance
- Maximum capacitance
- Maximum L or C ratio

Maximum L or C ratio is specified such that either Lmax/Lmin or Cmax/Cmin is to stay below that value. If exceeded, a warning is issued.

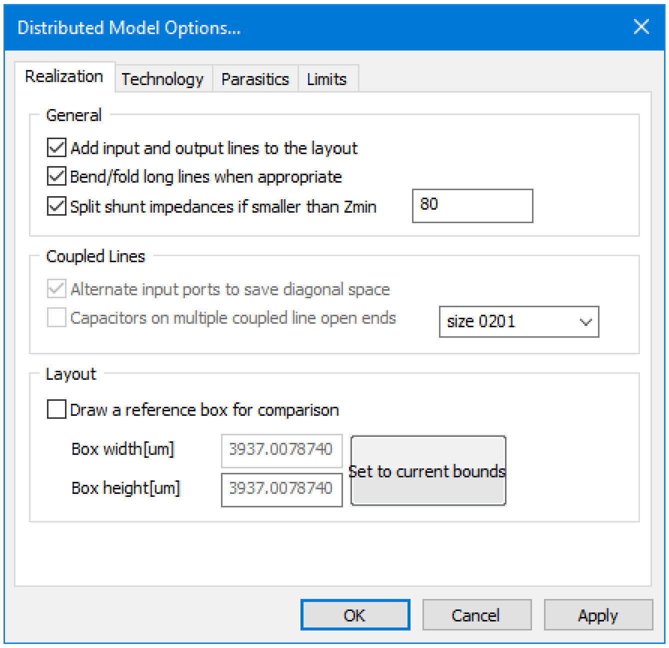
13.6.5. Distributed Model Options Dialog Box

The Distributed Model Options dialog box allows you to set up physical model options for microwave filters in general. To open the Distributed Model Options dialog box, click the **Design Options** button while designing a microwave filter type.

13.6.5.1. Distributed Model Options Realization Tab

You can edit general, coupled line, and layout options on the Distributed Model Options dialog box **Limits** tab. To view this tab, click the **Design Options** button in the main iFilter dialog box and then click the **Limits** tab.

You can scroll the mouse wheel in an edit box to increase/decrease the specified value. For example, doing so for Er steps through popular dielectric values, and doing so for the height parameter of a microstrip steps through 5, 10, 15, 20 ... board thicknesses. For microwave filters in general, the following dialog box displays.



This dialog box includes the following options:

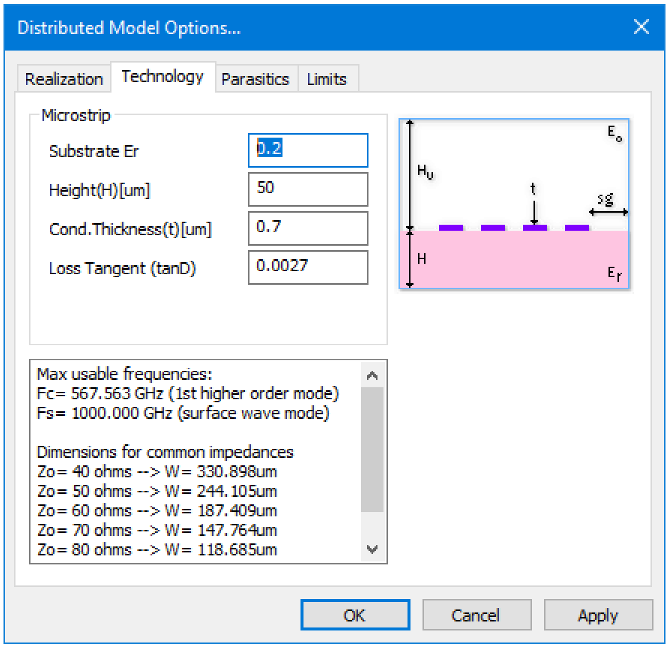
- **Add input and output lines to the layout** - adds lines at the filter input and output for the terminations. The extra line widths are calculated for the termination impedances, and a small length is used. You should select this check box, as you can use the extra lines to align the filter to the rest of the layout.
- **Bend/fold long lines when appropriate** - turns lines into traces (like MTRACE for MLIN) when the line is excessively long compared to the general layout of the filter. For stepped impedance resonator bandpass filter (SIR), this option is very useful.
- **Split shunt impedances if smaller than Zmin** - tells iFilter to split up shunt stub impedances (open or short circuited) into two identical elements and add them to the layout with a CROSS element, rather than a TEE element with one stub. You should do so, as most of the lowpass/bandpass structures have very low impedances. You can specify in the text box the threshold below which the splitting occurs.
- **Alternate input ports to save diagonal space** - alternates ports of edge coupled sections of SIR filters so that the layout stays along a horizontal axis. This practice saves diagonal layout space.
- **Auto rotate lines to save space when appropriate** - When you specify a rotation angle, this option rotates the filter to the specified degree, so that the layout can be realized at an angle.
- **Draw a reference box for comparison** - display a reference box around the layout. In the View Layout mode, when you change a filter parameter, the layout is recalculated and redrawn so you can view the resulting change.

13.6.5.2. Distributed Model Options Technology Tab

You can edit the technology (for example, microstrip or stripline) parameters on the Distributed Model Options dialog box **Technology** tab. To view this tab, click the **Design Options** button in the main iFilter dialog box and then click the **Technology** tab.

You can scroll the mouse wheel in an edit box to increase/decrease the specified value. For example, doing so for the **Substrate Er** parameter steps through an internal database of popular substrate dielectric constants. For any automatic selection of Er, the corresponding **Loss Tangent** for that substrate displays. Scrolling the mouse wheel for the Substrate

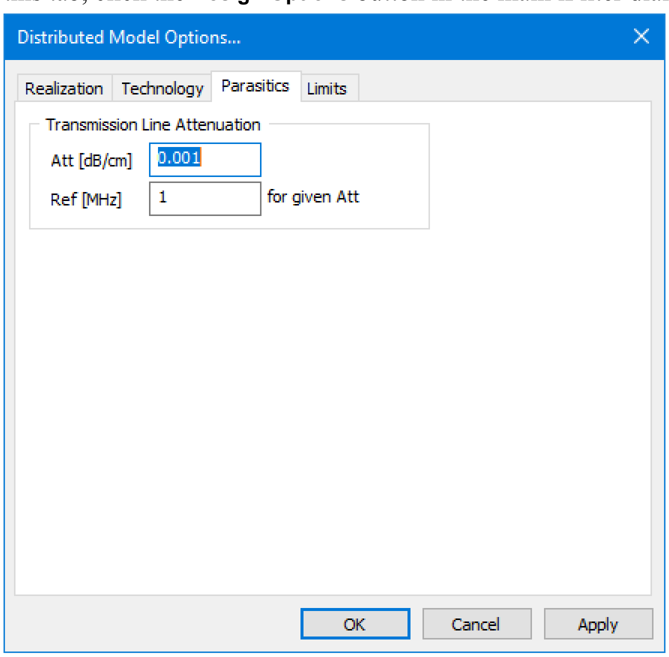
Height(H) steps through industry standard board thicknesses. To access this tab click the **Design Options** button in the main iFilter dialog box and then click the **Technology** tab on the Distributed Model Options dialog box.



The **Loss Tangent** is not used in calculating the dimensions; however it is included to correctly analyze the losses in the Microwave Office program.

13.6.5.3. Distributed Model Options Parasitics Tab

You can set loss parameters for distributed elements on the Distributed Model Options dialog box **Parasitics** tab. To view this tab, click the **Design Options** button in the main iFilter dialog box and then click the **Parasitics** tab.

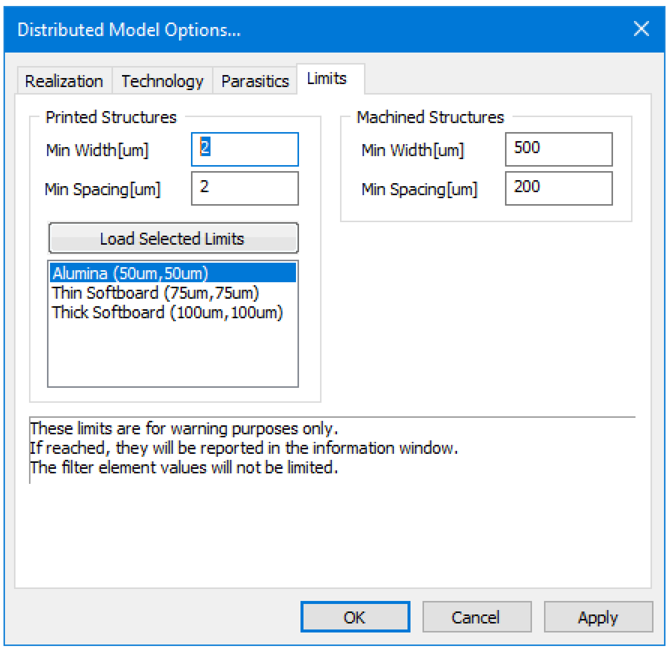


In this dialog box, you can specify the transmission line attenuation for distributed elements in Real analysis mode only. For RF and microwave filters, it is assumed that the attenuation of microstrip/stripline transmission lines is a linear function of frequency. For example, if a 10mm transmission line has 1dB attenuation at 1GHz, then it has 2dB attenuation at 2GHz. Because the filtering applications are diverse, a general loss factor is not suitable. The dB/cm approach is an old technique, but it works reasonably well. In this case, the attenuation is taking a linear function of frequency. So, 0.001dB at 1MHz increases to 0.010dB at 10MHz.

13.6.5.4. Distributed Model Options Limits Tab

You can edit physical element limits on the Distributed Model Options dialog box **Limits** tab. To view this tab, click the **Design Options** button in the main iFilter dialog box and then click the **Limits** tab.

For printed structures such as microstrip and stripline, the minimum width and minimum spacing (gap) are major manufacturing limits. When these limits are approached or exceeded, iFilter issues a warning by changing the color of the layout icon and elements in the layout, and by displaying a small warning icon next to the layout information entry. Some preset substrate type entries are listed. You can copy these values to **Min Width** and **Min Spacing** by selecting the desired entry and then clicking the **Load Selected Limits** button. Alternatively, you can double-click the desired entry or manually edit the option values.



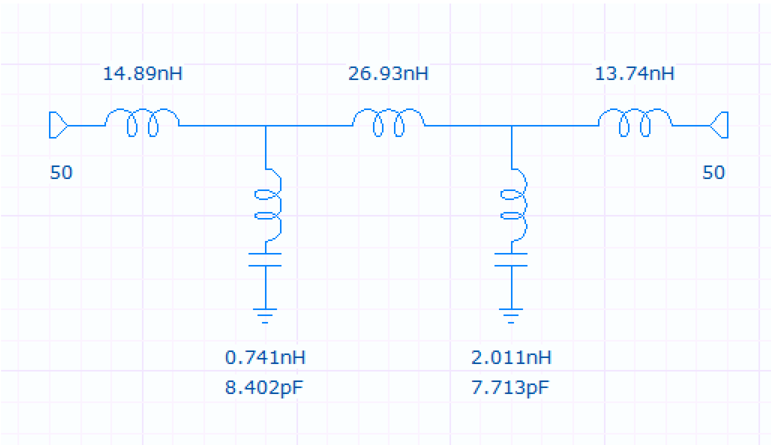
13.6.6. Lowpass Filters

Lowpass filters are designed to pass signals in a frequency range below a specified frequency, generally called the "cut-off" frequency. For filters that have an equi-ripple passband characteristic (for example, Chebyshev, Elliptic and Generalized Chebyshev), the cut-off frequency corresponds to the passband ripple corner. For other approximation types, it is the 3.011 dB (3-dB) corner frequency. The frequency range below cut-off frequency is called the "passband". The frequency range beyond the cut-off frequency is called the "stopband".

The passband of lowpass filters starts from f=0 Hz (DC). Within the passband the input impedance of the filter is very close to the source impedance, which is usually 50 ohms. In the stopband, impedance of the filter is no longer 50 ohms, and so rejects all the signals. In the stopband, filters are said to attenuate signals, more commonly called "rejection" than attenuation.

There is a limit to the stopband frequency range. For lumped element filters, due to the cavity of the housing, TEM and waveguide modes are excited at higher frequencies, so artificial passbands are observed. For example, a lowpass filter designed to have a 100MHz cut-off may show passbands beyond 1GHz. In these cases, a low order cleanup filter is cascaded to the main filter to suppress the artificial passbands. For distributed element filters, the response repeats itself due to the periodicity of electrical lengths. Therefore, beyond a certain frequency, lowpass filters behave like bandpass filters.

13.6.6.1. Lumped Element Lowpass Filter



Lumped element lowpass filters contains series inductors and shunt capacitors. For filters with finite TZs, SLC resonators replace the shunt capacitors. An elliptic lowpass filter is shown in the following figure.

Properties of lumped element filters are defined in "Lumped Model Options Dialog Box ".

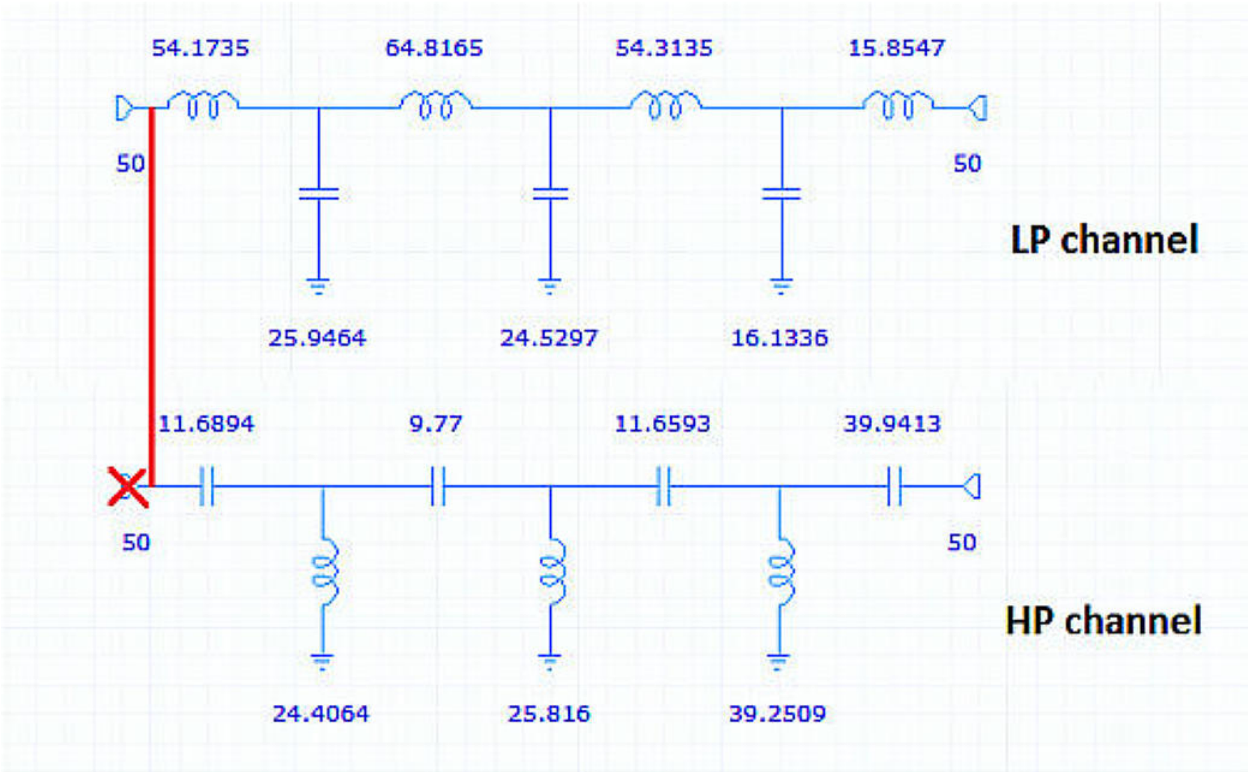
Typical Specifications

- **Approximation**: All available
- **Degree**: See standard ranges

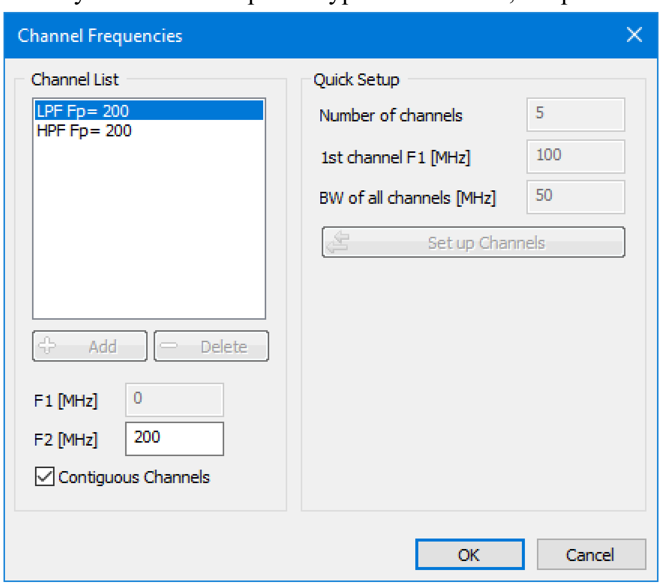
13.6.6.2. Lumped Lowpass/Highpass Diplexer

iFilter can design lowpass highpass diplexer using a "single terminated prototype" method. In this method, source impedance is first assumed ZERO ohms for both lowpass and highpass channels, and after combining the two channels in parallel, the source impedance is then set back to 50 ohms.

iFilter designs, analyzes and exports the diplexer channels as in the following figure. The two channels are normally connected at the source end and there is only one source termination. For display purposes, channels are displayed as if they have a source termination each.



When you select the diplexer type the first time, a diplexer setup dialog box displays.



The channel list on the left lists a lowpass and a highpass channel. The cutoff frequency can be set while editing the lowpass or the highpass channel. It is normally set to the same frequency for both channels, however iFilter provides the flexibility of setting them separately, so Fp can be slightly different to optimize the return loss.

This dialog box also allows you to edit bandpass multiplexer channels. The **Quick Setup** section provides controls for setting multiple channels. You can also specify the number of channels, the lower passband corner of the first channel, and the common bandwidth of all channels.

On the left of the main iFilter dialog box are channel access buttons. The first box shows the selected channel number. The next two buttons toggle between LP and HP channels. The **Edit** button displays the diplexer setup dialog box where you can edit the frequencies.

13.6.6.3. Stepped Impedance Lowpass Filter

Stepped impedance lowpass filters are very simple to construct. You can use almost any medium that creates a transmission line to make stepped lowpass filters. You can use a coaxial tube with varying inner rod diameters. iFilter primarily focuses on printed technology, therefore microstrip and stripline realizations are included.

A stepped lowpass filter is a series of low and high impedance lines. To make an analogy, a monotonic lumped element filter consists of series inductors and shunt capacitors. If series inductors are replaced by high impedance lines, and capacitors are replaced with low impedance lines, you can obtain a stepped lowpass filter. The line lengths are calculated from inductor and capacitor values and the response is approximated at the corner frequency.

There are a few drawbacks associated with using stepped impedance lines in filter structures:

- Line lengths must be iteratively adjusted, as the stepped capacitances affect the approximate equivalent circuit parameters. Various optimization routines are available in the literature. A simple approach taken in iFilter is to design the filter by taking the shift in performance into account from the previous iteration. This is a trivial yet effective solution with little need to adjust return loss after design.
- The input impedance of these filters never reaches a ZERO or INFINITE impedance and they never show full reflection. As a result, these filters do not possess very good stopband attenuation.
- A drawback (or advantage if used properly), is the recurring passband. It may be shifted by adjusting the impedances to suppress undesired harmonics or spurious regions.

This filter type provides two options: **Set Z, varying lengths** and **Same Length, varying Z's**.

The **Set Z, varying lengths** option uses lumped element prototypes as a basis, and replaces the prototype elements with transmission lines specified as low and high impedances. Line lengths are calculated for each element, to achieve the best approximation in the passband. For some element values the line lengths may not be realizable for the required impedances. If this occurs, the specified impedances are changed until the approximation yields a positive length for that element. You should specify **Low Zo** as low as possible to obtain a better stopband rejection.

The **Same Length, varying Z's** option uses commensurate line synthesis. The transmission lines all have the same specified electrical length **ElecLng** (EL) at the passband corner, **Fp**. EL controls two important aspects of the filter: spurious performance and impedance ratio. As with every distributed filter, there are inevitable spurious passbands. For this type of filter, the spurious passband occurs at Fp * (180-EL)/EL. So, if EL=45 degrees, a spurious passband occurs starting at 3*Fp. When EL is reduced to 30 degrees, the spurious passband corner moves to 5*Fp. So a lower EL moves the undesired spurious passband away from the intended passband. EL also controls the high impedance/low impedance ratio of the filter. The higher the EL, the lower the impedance ratio making the topology more practical to build. Therefore, a trade-off between practicality and spurious response should be decided while adjusting EL.

Typical Specifications

- **Approximation**: Monotonic
- **Degree**: See standard monotonic ranges
- **Low Zo** 15 to 50 ohms (must be realizable) – for Set-Z option only. Specify as low as possible.
- **High Zo**: 50 to 250 ohms (must be realizable) – for Set-Z option only.
- **ElecLng**: 10 to 60 degrees – for Same-Length option only.

Tuning and Optimization

- L_v(n) - are the element line lengths. Usually, the step capacitance between impedance transitions slightly affects the passband corner and the return loss. Some tuning may be required for the lengths.

13.6.6.4. Distributed Stubs Filter

This filter uses lumped element prototypes as a design basis and replaces prototype elements with transmission lines. The distributed stubs lowpass filter consists of transmission lines separated by open-circuited shunt stubs. Transmission line impedances are intended to be user-controlled; so the Line Zo parameter is available for editing. In most cases, however, only the input and output transmission lines can give valid line lengths. For internal lines, the impedances are increased to obtain a valid line length approximation. A higher Line Zo yields better return loss in the passband.

Open-circuited shunt stub impedances are independent of the Line Zo specification. Their impedances and lengths are calculated from the lowpass prototype elements.

Typical Specifications

Approximation: Monotonic

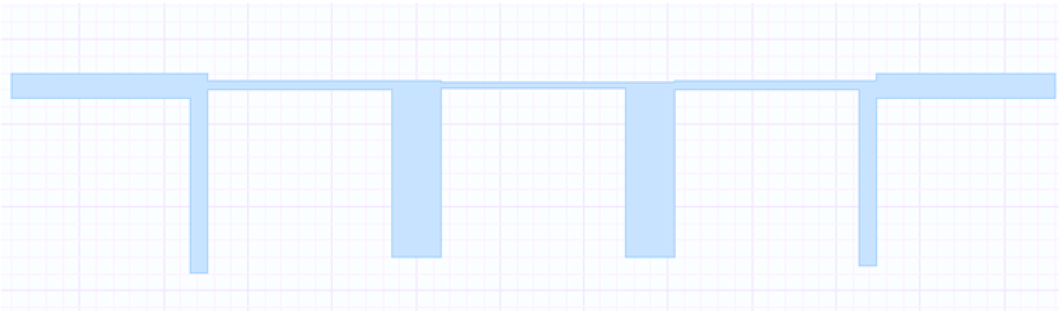
Degree: See standard monotonic ranges

ElecLng: 30 to 150 ohms (must be realizable)

Tuning and Optimization

L_v(n) - are the element line lengths. Usually, the step capacitance between impedance transitions slightly affects the passband corner and the return loss. Some tuning may be required for the lengths.

13.6.6.5. Optimum Distributed Lowpass Filter



Distributed element lowpass filters may contain short-circuited stubs in the series arms and open-circuited (open ended) stubs in the shunt arms. In planar structures like microstrip, short-circuited stubs in the series arm are not realizable. One solution is to convert these stubs into practical open ended shunt stubs using Kuroda transformations, so a 5th order lowpass filter consists of five shunt stubs separated by four transmission lines.

An optimum distributed lowpass filter is much the same in appearance, but with one important difference in the response. The transmission lines in the optimum filter are not obtained by Kuroda transformations, but rather synthesized into the transfer function during the design. Each transmission line therefore adds a transmission zero to the S21 response. Thus, the same 5 stubs + 4 line filter above will have 9th order response for the optimum case, with a much sharper stopband selectivity.

This filter type lets the electrical length (EL) at passband corner be specified. As with the stepped impedance lowpass filter type, the EL controls the spurious passband frequency and impedance ratio of the filter. A trade off might be necessary while setting EL, which is easy to observe in the main IFilter dialog box.

Typical Specifications

Approximation: Chebyshev only with fixed 26dB, 20dB or 16dB return loss.

Degree: 3 to 19

ElecLng: 18 to 85 degrees

Tuning and Optimization

L_v(n) - are the element line lengths. Slight tuning may be required for the lengths for accurate passband corner and return loss.

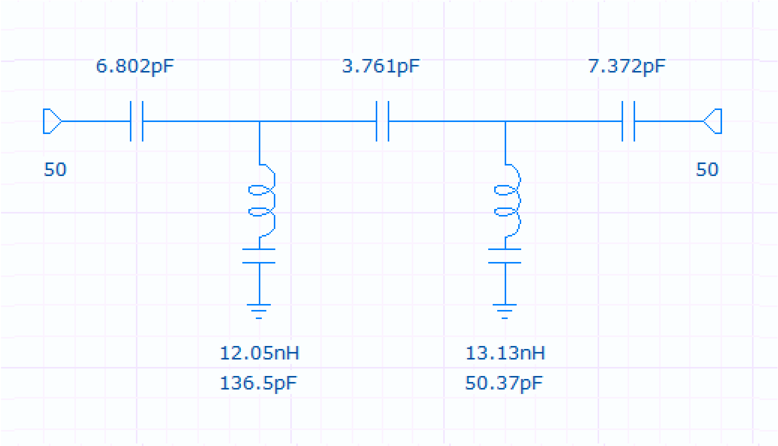
13.6.7. Highpass Filters

Highpass Filters are designed to pass signals in a frequency range above a specified frequency, generally called the "cut-off" frequency. For filters that have an equi-ripple passband characteristic (for example, Chebyshev, Elliptic and Generalized Chebyshev), the cut-off frequency corresponds to the passband ripple corner. For other approximation types it is the 3.011 dB (3-dB) corner frequency.

There is an upper frequency limit. For lumped element filters, the upper limit is dictated by the performance of the lumped elements, which are limited by loss and self-resonance frequencies. For distributed element filters, the response repeats itself due to the periodicity of electrical lengths. Therefore, beyond a certain frequency, highpass filters behave like bandpass filters and they attenuate beyond a certain frequency. Printed type distributed filters (microstrip and stripline) also have a dielectric loss that limits the maximum frequency for which the performance closes the specification.

13.6.7.1. Lumped Element Highpass Filter

Lumped element highpass filters contains series capacitors and shunt inductors. For filters with finite TZs, SLC resonators replace the shunt inductors. An elliptic highpass filter is shown in the following figure.



The realization of lumped element filters is defined in "Lumped Model Options Dialog Box ".

Typical Specifications

- **Approximation**: All available
- **Degree**: See standard ranges

13.6.7.2. Shunt Stub Highpass Filter

A shunt stub highpass filter contains shunt short-circuited stubs and connecting transmission lines. There are three options available for shunt stub highpass filters.

The first option (**1/4wave lines + 1/4wave stubs**) is a combination of quarterwave length stubs separated by quarterwave length transmission lines. The second option (**1/4wave lines + 1/4wave stubs (equal)**) is a subtle variation of the first, as the stubs are made equal using exact circuit transformations. The response is the same, and the stub impedances are more realizable in most practical cases.

The last option (**1/4wave lines + 1/2wave stubs**) is a combination of halfwave length stubs separated by quarterwave length transmission lines. This option has much sharper response than the first two options as it generates a transmission zero below passband corner. However, it also generates a lowpass response around DC.

Shunt stub highpass filters behave exactly like bandpass filters. The upper passband corner of highpass filters does not extend to infinity, but rather a finite frequency. Electrical length (EL) controls the width of the highpass filter.

By specifying EL at Fp, a bandpass filter can be assumed to have a passband of Fp to Fo+2*(Fo-Fp), where the transmission lines are 90 degrees long at Fo. You can deduce the following:

Fo = Fp * 90/EL

For example, a shunt stub highpass filter designed at 4 GHz with 60 degree long lines has a passband of 4 to 8 GHz with Fo=6 GHz.

If the EL is specified high, the resulting passband is narrow and the shunt stubs become very low impedance, necessitating wide strips for planar realization. These filters are therefore more suitable with wide passbands (EL < 70 degrees).

Typical Specifications

- **Approximation**: Monotonic
- **Degree**: See standard monotonic ranges (with limitation of 3 to 19 range)
- **ElecLng**: 50 to 85 degrees

Tuning and Optimization

- L_v(n) - are the element line lengths. Slight tuning may be required for the lengths for accurate passband corner and return loss.

13.6.7.3. Optimum Distributed Highpass Filter

An optimum distributed highpass filter is a subtle variation of a shunt stub highpass filter. The optimum type is obtained by exact synthesis of (n-1) transmission lines and a shunt stub, which is modified using Kuroda transformations. The stub lengths are also set to be equal during the transformation stage. The outcome is a filter with perfect return loss and identical shunt elements.

All elements of the filter have specified electrical length (EL) at the passband corner. Like the shunt stub highpass filter, the EL controls the width of the passband and the level of impedances. The optimum distributed highpass filter is suitable for moderate to wide passbands (for example, from EL=40 to 70-75 degrees).

Typical Specifications

- **Approximation**: Chebyshev only with fixed 26dB, 20dB or 16dB return loss.
- **Degree**: 3 to 14
- **ElecLng**: 9 to 85 degrees (extreme specs result in impractical impedances)

Tuning and Optimization

- L_v(n) - are the element line lengths. Slight tuning may be required for the lengths for accurate passband corner and return loss.

13.6.8. Bandpass Filters

Bandpass filters are the most common RF and microwave filters. They are designed to pass signals within a certain frequency range. The upper and lower limits of the passband are generally called the lower and upper cut-off frequencies. For filters that have equi-ripple passband characteristics (Chebyshev, Elliptic, and Generalized Chebyshev), cut-off frequency corresponds to the passband ripple corner. For other approximation types it is the 3.011 dB (3-dB) corner frequency.

The passband of bandpass filters is also called bandwidth of the filter and often denoted as BW. The middle frequency is termed as center frequency and often denoted as Fo.

An important property of bandpass filters is the insertion loss. The insertion loss is the minimum achievable loss of the filter in the passband, mostly occurring at Fo. Insertion loss plays a great role in receivers as it adds to the system noise figure, reducing the sensitivity. In transmitters, it causes dissipation in the filter and a reduction in power transmitted. For a 10W transmitter, 1-dB insertion loss corresponds to 20% reduction in power (2W). Not only is the delivered power 2W less, but the dissipation of such high power increases the temperature of the filter significantly.

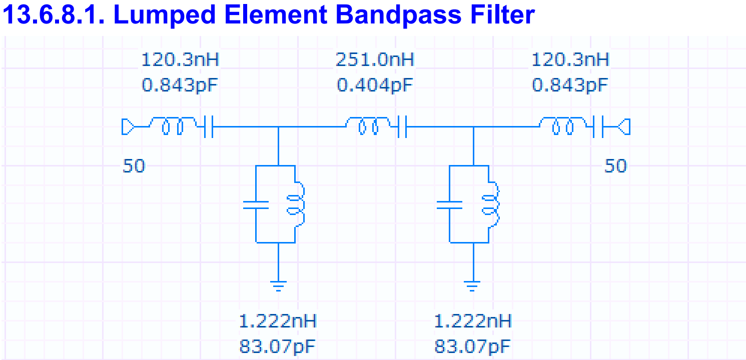
Insertion loss depends on the bandwidth of the filter, unloaded Q of the resonators, and the degree of the filter. If a filter is narrowband, it becomes lossy, so high Q elements are needed to minimize the filter loss. A simple yet accurate expression for bandpass filter insertion loss is given in "Design Utilities Midband IL (Midband Insertion Loss) Tab".

Most bandpass filters are designed for relatively narrow bandwidths. For lumped element type topologies, filters derived from standard LP prototypes suffer a wide element value range. A 7th degree, 10% standard filter designed at 500MHz yields an inductor range of 0.97nH to 278nH (1:284 impedance ratio) which requires 1-turn and 20-turn coils at the same time. The same filter also has a 1:284 impedance ratio for the capacitor values. Also, filters derived from inverter prototypes yield a more realizable element parameter range. For the same specifications, an inverter-based filter turned into inductively coupled capacitive pi sections uses only 1 value of inductors (126nH) and the capacitance range is limited to 0.15 to 1.45pF (1:9.8 impedance ratio).

For distributed element bandpass filters there are many options based on the available technology. If a microstrip or stripline technology is used, and if area permits, you can design edge coupled or stepped impedance resonator filters.

Hairpin and interdigital filters offer great advantages on microstrip. For machined structures, combline is ideal for machining and tuning.

13.6.8.1. Lumped Element Bandpass Filter



A lumped element bandpass filter is created by applying frequency transformation to the lowpass prototype filter.

For monotonic filters, this corresponds to SLC resonators on the series arm and PLC resonators on the shunt arms. For filters with finite TZs, shunt arms contain LC-quad elements. In iFilter, quad elements are replaced by SLC resonators on the shunt arm.

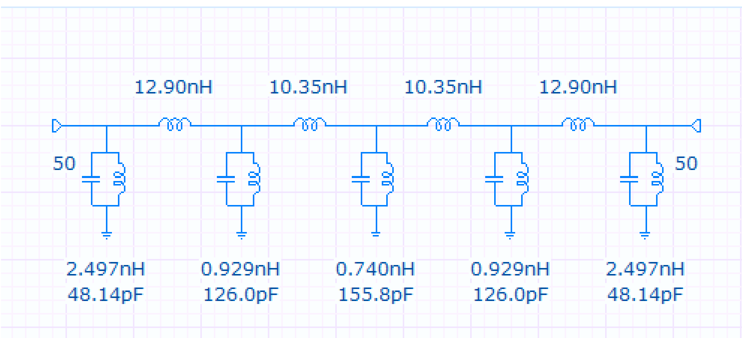
This type of filter gives accurate response for any bandwidth, however, the element parameter magnitudes can be widely spread. The ratio of maximum to minimum inductance or capacitance can easily reach beyond 100. In that case, you cannot use components from the same part family, or the parasitics effects are difficult to tune out.

Realization of lumped element filters is defined in "Lumped Model Options Dialog Box ".

Typical Specifications

- **Approximation**: Monotonic
- **Degree**: See standard monotonic ranges
- **BW**: 0 to 200% (extreme specs result in impractical impedances)

13.6.8.2. Narrowband Lumped Element Filter



A narrowband lumped element filter is derived from inverter prototypes. The main difference between this filter and conventional bandpass filters is the use of series arm components and their parameter range. For conventional filters, the series arm consists of LC-resonators. For narrowband filters, you can approximate the series arms (coupling elements) with single inductors or capacitors. The other difference is the spread in parameter values. For narrowband filters, the spread in parameter values is much less than that found with the conventional type.

There are six options in iFilter for narrowband lumped element bandpass filters. The first three are Inductively coupled options. These contain a series inductor as the coupling elements. Inductively coupled type filters are quasi-lowpass structures, where the stopband attenuation is steeper on the high end of the filter. The other three options use Capacitively coupled shunt sections. These contain a series capacitor as the coupling elements. Capacitively coupled type filters are quasi-highpass structures, where the stopband attenuation is steeper on the low end of the filter. When bandwidth gets wider, the high end attenuation almost disappears for low order filters.

The first option is the generic **Inductively Coupled** type. It is a popular narrowband filter, however as it contains many inductive components, it tends to be lossy. With proper setting of parasitics and unloaded Qs, this type can simulate filters of quasi-lowpass nature, such as combline filters. For more information, see "Arbitrary Narrowband Filter Simulation Example".

The second option is the **Inductive (identical Shunt C)** option where all the shunt capacitors are equated to the same value. This is an attractive feature for applications that require a narrow capacitance range. Tunable filters are prime candidates, where they can be designed with identical tuning diodes using this topology.

The third option is the **Inductively coupled Cap-Pi's** where the resonating shunt capacitances of the original inductively coupled type are now replaced by capacitive Pi-sections. You can use this topology to design tubular filters, which are not included in this version of iFilter. This topology has the minimal element value spread of the inductively coupled filters.

The fourth option is the generic **Capacitively coupled** type. It is another popular narrowband filter, and unlike the inductively coupled type, it does not suffer from insertion loss, as it contains mainly high-Q capacitors.

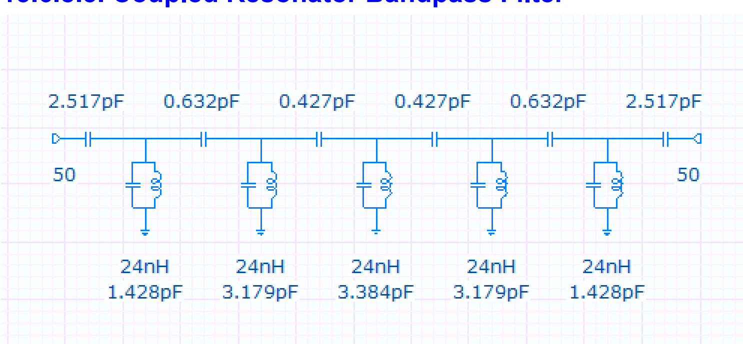
The fifth option is the **Capacitive (identical shunt L)** option where all of the shunt inductors are engineered to be the same value. Making all inductors share the same value uses an iterative technique. This process yields inductor values close enough in magnitude that you can use the same coil structure and slightly tune them.

The sixth option is the **Alternating Ind/Cap** option where the coupling elements alternate between inductor and capacitor. The response is fairly symmetric, not skewing the stopband performance.

Typical Specifications

- **Approximation**: Monotonic
- **Degree**: See standard monotonic ranges (with limitation of 2 to 25 range)
- **BW**: 0 to 50% (wider bandwidths suffer mismatched return loss)

13.6.8.3. Coupled Resonator Bandpass Filter

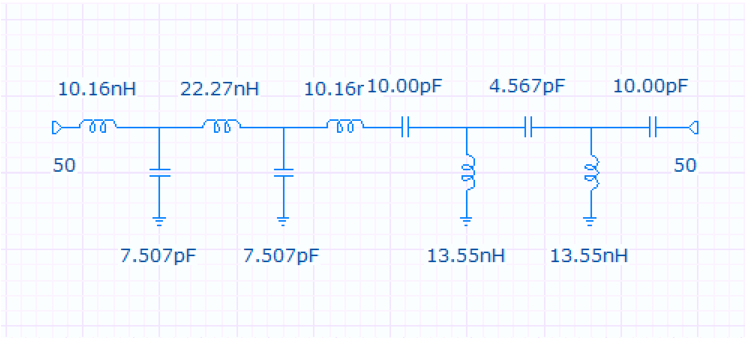


The coupled resonator filter is a special form of a Capacitively Coupled Narrowband Filter with equal shunt inductors. It has the advantage of using series input and output capacitors, making it easy to connect to terminations.

Typical Specifications

- **Approximation**: Monotonic
- **Degree**: See standard monotonic ranges (with limitation of 1 to 25 range)
- **BW**: 0 to 50% (wider bandwidths suffer mismatched return loss)
- **Lshunt**: Any suitable range

13.6.8.4. Wideband Lumped Element LP+HP Filter



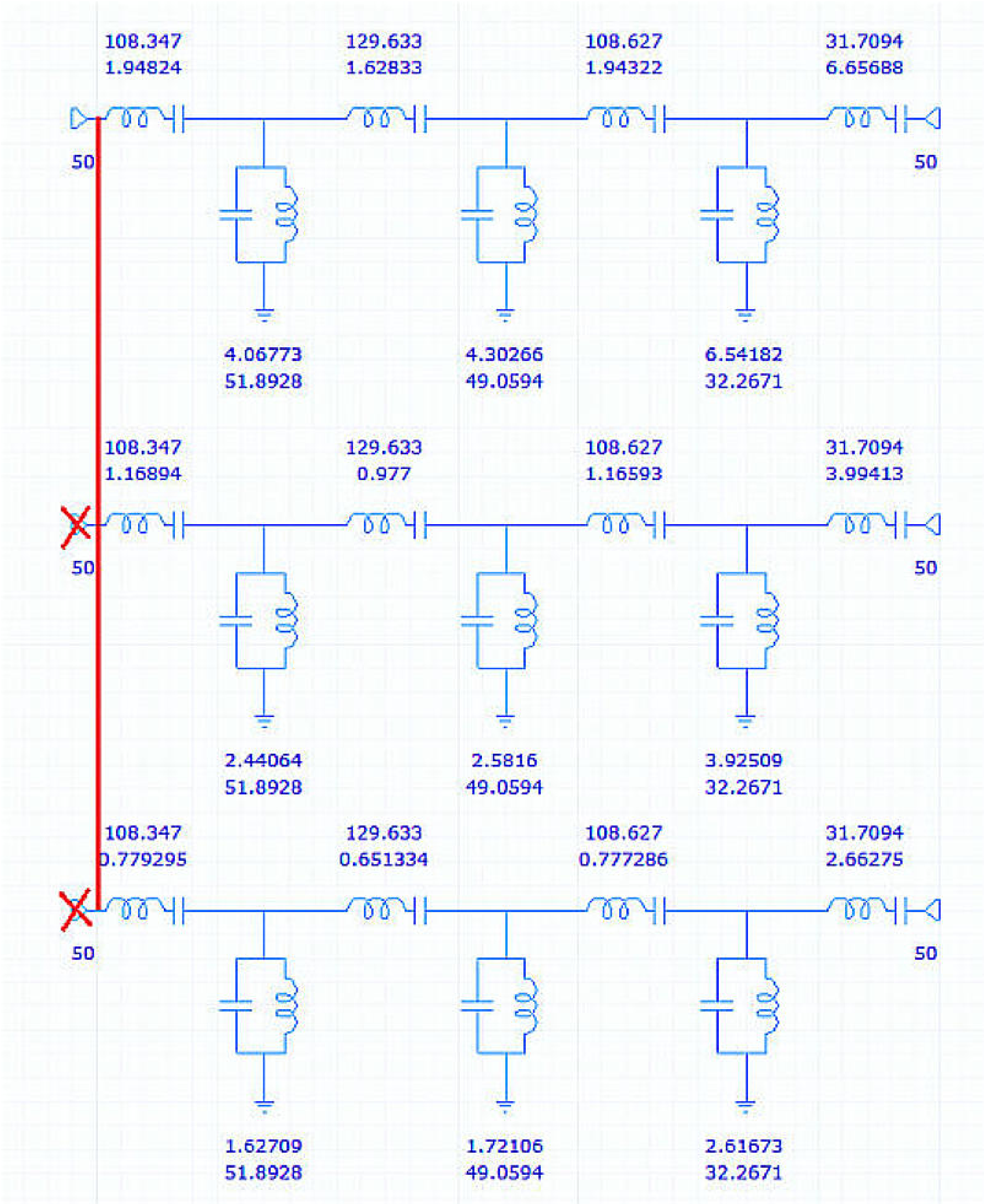
A wideband lumped element filter is obtained by cascading lowpass and highpass filters. For wide bandwidths, the two filters do not interact, so tuning the passband corners is very easy, as the lowpass side controls the upper frequency corner and the highpass side controls the lower frequency corner.

Typical Specifications

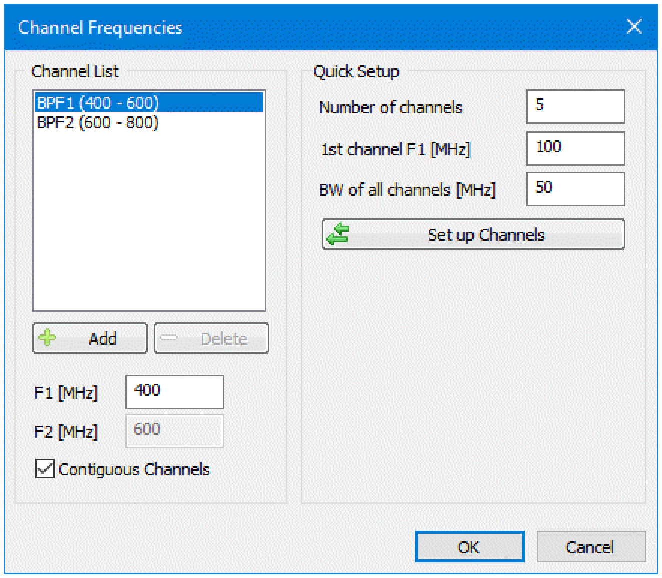
- **Approximation**: Monotonic
- **Degree**: See standard monotonic ranges (with limitation of 1 to 25 range)
- **BW**: 20 to 200% (narrower bandwidths suffer interactions)

13.6.8.5. Lumped Bandpass Multiplexer

iFilter can design lowpass bandpass diplexers using a "single terminated prototype" method. In this method, source impedance is first assumed ZERO ohms for both multiplexer channels, and after combining all channels in parallel, the source impedance is then set back to 50 ohms. iFilter designs, analyzes, and exports the diplexer channels as in the following figure. The channels are normally connected at the source end and there is only one source termination, however, for display purposes, channels are displayed as if they have a source termination each.



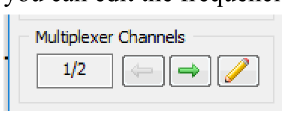
When you select the multiplexer type the first time, a multiplexer setup dialog box displays.



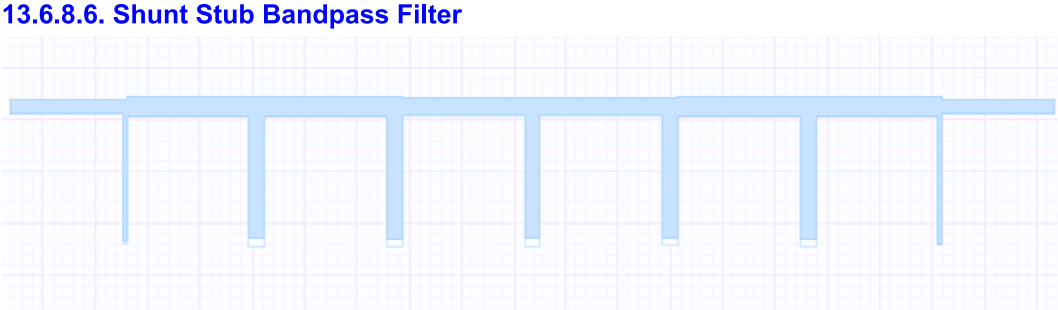
The channel list on the left lists bandpass channels. The lower passband cutoff frequencies can be set while editing each channel. The cutoff frequency is normally set to the same frequency for adjacent channels, however iFilter provides the flexibility of setting them separately when you select the **Contiguous Channels** check box, so frequency corners can be slightly different to optimize the return loss.

The **Quick Setup** section provides controls for setting multiple channels. You can also specify the number of channels, the lower passband corner of the first channel, and the common bandwidth of all channels.

On the left of the main iFilter dialog box are channel access buttons. The first box shows the selected channel number. The next two buttons toggle between LP and HP channels. The **Edit** button displays the diplexer setup dialog box where you can edit the frequencies.



13.6.8.6. Shunt Stub Bandpass Filter



A shunt stub bandpass filter is the same as a shunt stub highpass filter with the specification reformatted. For highpass filters, Fp and EL are defined. For bandpass filters, Fo and BW are defined. The following relations can therefore be established:

Fo = Fp * 90/EL

BW = 2*(Fo-Fp)

The comment for the shunt stub highpass filter is also valid for the shunt stub bandpass filter. The passband of the bandpass filter is controlled by BW rather than EL.

Typical Specifications

- **Approximation**: Monotonic
- **Degree**: See standard monotonic ranges (with limitation of 3 to 19 range)
- **BW**: 0 to 200% (extreme specs result in impractical impedances)

Tuning and Optimization

L_v(n) - are the element line lengths. Slight tuning may be required for the lengths for accurate passband corner and return loss.

13.6.8.7. Optimum Distributed Bandpass Filter

An optimum distributed bandpass filter is the same as an optimum distributed highpass filter with the specifications reformatted. For highpass filters, Fp and EL are defined. For bandpass filters, Fo and BW are defined. The following relations can therefore be established:

Fo = Fp * 90/EL

BW = 2*(Fo-Fp)

The comment for the optimum distributed highpass filter is also valid for the optimum distributed bandpass filter. The passband of the bandpass filter is controlled by BW rather than EL.

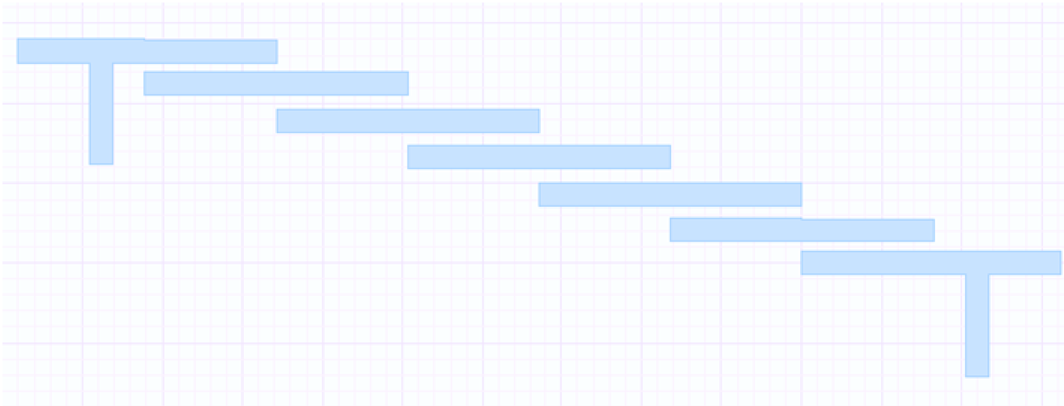
Typical Specifications

- **Approximation**: Chebyshev only with fixed 26dB, 20dB or 16dB return loss.
- **Degree**: 3 to 14
- **BW**: 10 to 180% (extreme specs result in impractical impedances)

Tuning and Optimization

- L_v(n) - are the element line lengths. Slight tuning may be required for the lengths for accurate passband corner and return loss.

13.6.8.8. Edge Coupled Bandpass Filter (Parallel Coupled Line Filter)



This filter type is also known as Half-Wavelength Parallel Coupled Line Filter, because it uses half-wavelength resonators. The term "edge coupled" is used in iFilter, as the commonest media are microstrip and stripline, where the resonators are coupled through their edges. For Suspended Stripline, or Broadside Coupled Stripline, the term "edge coupled" is not literally correct.

Because of its unique longitudinal shape, it is one of the most popular bandpass filters in moderate bandwidth microwave applications. Another advantage of these filters is that they do not require grounding. On the negative side, two things are worth considering when designing edge coupled filters:

The outermost coupled sections (next to the terminations) require significant coupling, so the spacing is normally tighter than that found with the internal resonator sections. For higher fractional (or percentage) bandwidths, more coupling is required, which necessitates that the resonators be even closer. Due to manufacturing limitations, the minimum gaps between lines must be followed. Naturally, this limits the fractional bandwidths that can be obtained. You can select thicker substrates to obtain wider spacing, but it is at the expense of increasing the cost and size.

The second disadvantage is the spurious frequency response. Theoretically, this topology possesses spurious passbands at odd multiples of the desired center frequency, Fo. If an ideal coupled line filter is designed at 2GHz, it naturally passes 3*Fo, 5*Fo components at 6GHz and 10GHz. For a homogeneous medium like stripline, actual spurious passband content is as calculated in ideal coupled line cases. Microstrip, although it is far more popular than stripline, is not a homogeneous medium, so the even and odd mode mismatch results in the 2*Fo spurious passband emerging. The same 2GHz filter, therefore, if built on a microstrip, has a passband of approximately 4GHz. To counter this, special techniques exist such as wiggly spacings between resonators, and lengthening/shortening resonators beyond the coupled region. In the current version of iFilter only length-adjustment is available.

Edge coupled filters are available in microstrip and stripline.

The microstrip transmission medium exhibits unequal even and odd mode effective dielectric constants. This results in uneven guided wavelengths which in turn cause unwanted spurious passband about 2*Fo. To counter this phenomenon, there are several design techniques such as making the coupled lines "tapered" or "wiggly" or defected ground structures. No such guidance data for wiggly lines has been available for all substrates and thickness. You should design the filter using the technique provided first, then by exporting it into the EM simulator, you should fine-tune it for better spurious performance. It is time consuming to concurrently design and tune a microstrip edge coupled filter structure.

The stripline medium does present the same problem, so it is relatively easier to work with stripline filters. These filters also have a drawback, however. For the same specifications, the first and last resonator sections tend to be more closely spaced for stripline. One solution is to use tapped input/output sections which replace the two coupled line sections with open stubs.

You can specify the source and load impedances as other than 50 ohms, in which case the even and odd mode impedances of the nearest coupled line sections are adjusted accordingly for impedance matching.

There are three options for edge coupled filters: Impedance Controlled, Standard, and Tapped input/output.

For the **Impedance Controlled** option, the **Reson Zo** parameter controls the internal impedance level of the coupled resonators. If **Reson Zo** is specified as close to 50 ohms, the coupled line impedances and line widths are comparable to terminating lines. If a higher impedance is used for **Reson Zo**, the coupled sections have higher even and odd mode impedances, and the lines are narrower. For tight couplings, a high **Reson Zo** is recommended, as it results in slightly higher spacings. You should note, however, that the insertion loss increases due to narrower lines. This option works for moderate bandwidths.

The second option is the **Standard** textbook option where no impedance control is available. This option can achieve slightly higher bandwidths than the **Impedance Controlled** option.

The third option is the **Tapped input/output** option. This option is a variation of the **Impedance Controlled** type. The input/output coupled sections, which have tight spacings, are replaced with an open-circuited shunt stub and a short transmission line. As a result, a more practical circuit is obtained. Another advantage of the **Tapped input/output** type is the improved spurious suppression. As the tapping sections do not conform to commensurate (identical) lengths of the resonators, the spurious passband may occur at arbitrary frequencies depending on the phases and impedances of lines. The two associated drawbacks of the tapped option are the availability of achievable bandwidths and the imperfect return loss. For wide bandwidths, the tapping section cannot simply provide the replaced coupling, therefore you should slightly tune internal couplings. You should also slightly tune Return loss due to the finite capability of the tapping. The **Tapped input/output** option is optimized for 20dB return loss (0.0436dB passband ripple), so using other passband ripple values may not result in a good match.

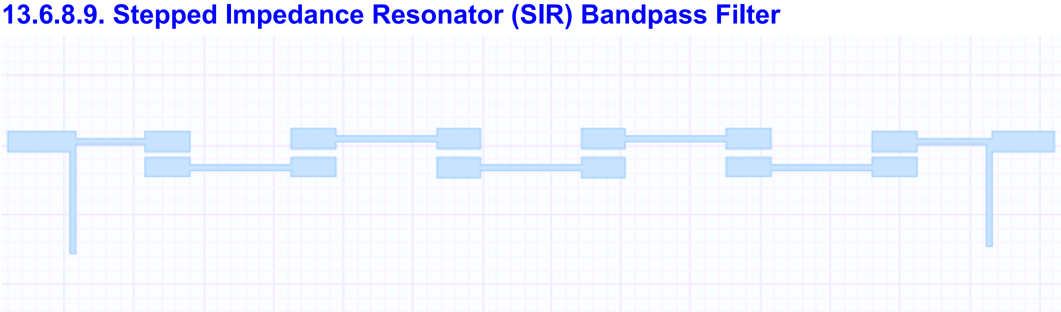
Typical Specifications

- **Approximation**: Monotonic (for **Impedance Controlled** and **Standard**) and Chebyshev only (for **Tapped input/output**)
- **Degree**: 3 to 15
- **BW**: no limitation, but reasonable performance for up to 40%
- **Resonator Zo**: 30 to 120 ohms (must be realizable) – yields thinner lines with increasing Zo

Tuning and Optimization

- L_v(n) - are the lengths of various line sections. Tuning may be required to set the response to the desired center frequency. For microstrip, you should tune the length parameter which is initially set to 0.000254mm (0.01mil). Negative values of this parameter suppress 2*Fo spurious passband, however the coupled line lengths should be slightly tuned to move the response back to the desired center frequency.

13.6.8.9. Stepped Impedance Resonator (SIR) Bandpass Filter



Stepped impedance resonator bandpass filters look like edge-coupled bandpass filters with extra transmission lines between the coupled sections. They offer the following advantages:

- They do not suffer 2*Fo spurious response as half-wave edge coupled filters. The spurious response can be shifted by changing the transmission line impedance.
- Shifting harmonic response at 2*Fo is especially useful in oscillator and amplifier designs.
- The connecting lines can be made thin enough to bend in any shape, which provides more layout options. The layout can be made very compact.
- The input and output ports do not have to be placed along the same axis.

The spurious passband of edge coupled filters is not particularly controllable. It appears at odd multiples, for example, 3*Fo and 5*Fo. In addition, for microstrip filters, the even multiples come into effect. The SIR bandpass filter is a slight variation of edge-coupled filters and was created to address this issue. By converting part of the coupled sections into simple transmission lines, the spurious passband can be moved further away from the desired passband. The inserted transmission lines can then be bent for improved realization, resulting in a compact filter. The drawback is that the available fractional bandwidth is narrower than comparable edge-coupled filters. The application is limited to 30% or less bandwidths in practice.

The stepped impedance resonator filter allows editing of the **Line Zo** (the transmission line pieces). **Line Zo** controls the location of spurious passband. Specifying a high value for **Line Zo** causes the spurious passbands to move away from the desired center.

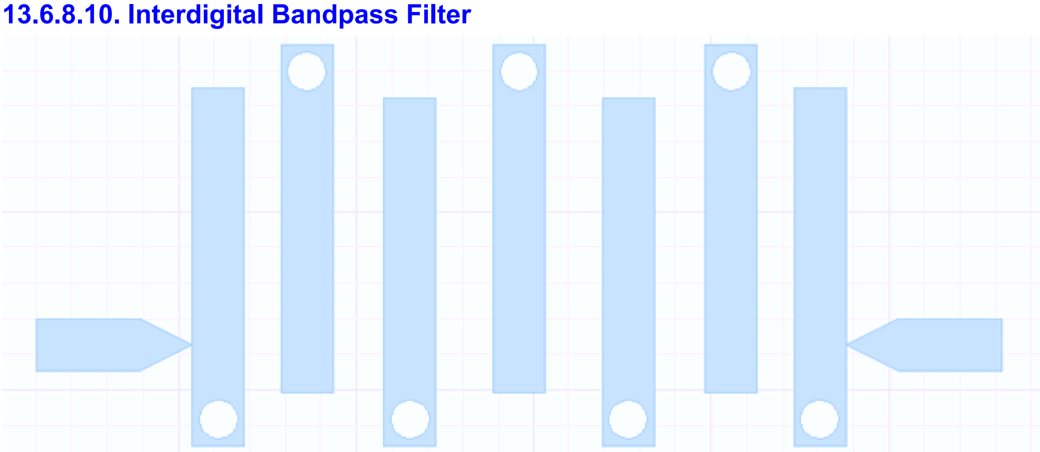
Typical Specifications

- **Approximation**: Monotonic
- **Degree**: See standard monotonic ranges (with limitation of 2 to 15 range)
- **BW**: no limitation but reasonable performance for up to 25-30%
- **Resonator Zo**: 30 to 150 ohms (must be realizable)

Tuning and Optimization

- L_v(n) - are the lengths of various line sections. Tuning may be required to set the response to the desired center frequency.

13.6.8.10. Interdigital Bandpass Filter



Interdigital bandpass filters consist of tightly spaced vertically oriented resonators, so they offer significant size advantage over most other microwave filters. Originally intended for round rod and rectangular bar technologies, interdigital filters are now used in microstrip more than other filters, due to their simplicity in realization.

Interdigital resonators have fixed 90 degree electrical lengths. They are open-ended on one end and short-circuited on the other. In microstrip or stripline, a short-circuit is provided by vias. In rectangular bars and round rods, the short-circuited end is connected to the housing.

Interdigital filters can be approximated more accurately in homogeneous media such as stripline. In iFilter, there are various options depending on the selected technology.

For microstrip, a non-homogeneous medium, tapped and uniform width line options are available. iFilter uses EM-generated data for calculating spacing between resonators. The resulting structure still needs tuning, mostly tapping lengths and/or the outermost spacings. For other technologies, non-uniform widths are also available.

Typical Specifications

- **Approximation**: Monotonic
- **Degree**: See standard monotonic ranges (with limitation of 16)
- **BW**: no limitation but reasonable performance for up to 50%

Tuning and Optimization

- L_v(n) - are the lengths of various line sections. Tuning may be required to set the response to the desired center frequency.
- LTOT - is the length of the resonators. Increasing/decreasing it moves the center frequency of the filter down/up respectively.
- LBot - is the tapping length at the input and output measured from the end of the resonator. For the input side, this is the bottom of the resonator. For the output side, it is the top or bottom of the resonator if the number of resonators is an even or an odd number.
- S(n)_v(n) - are the spacings between resonators. Tuning may be required to set correct passband width.

13.6.8.11. Combline Bandpass Filter

Combline bandpass filters exploit the interdigital filter idea, where all the resonators are coupled together in a small form factor. In addition, combline resonators are adjusted to shorter lengths (down to 30 deg), and the resonance is obtained by adding a tuning capacitor. For planar filters, the capacitor is simply a lumped element. For machined structures like round rods and rectangular bars, capacitance is obtained by tuning rods.

Combline filters possess a topology similar to interdigital filters, however the orientation of combline resonators are all the same (all are short-circuited at the same side). This provides the advantage of tuning the filter at the open ends, which are on the same side of the housing. For planar filters like microstrip and stripline, this is not a big advantage, but for machined structures it is.

Combline filter resonator couplings are inductive, so the response is of quasi-lowpass nature (the selectivity obtained on the upper side of the passband is much sharper than on the lower side).

Typical Specifications

- **Approximation**: Monotonic
- **Degree**: See standard monotonic ranges (with limitation for 3-16 range)
- **BW**: no limitation, but reasonable performance for up to 50%
- **ElecLng**: 30 to 85 degrees

Tuning and Optimization

- L_v(n) - are the lengths of various line sections. Tuning may be required to set the response to the desired center frequency.
- S(n)_v(n) - are the spacings between resonators. Tuning may be required to set the correct passband width.

13.6.8.12. Hairpin Bandpass Filter

Together with interdigital and combline filters, hairpin filters offer the smallest size for a given number of resonators. Hairpin filters may be thought of as a variation of edge coupled filters. The difference is that the short line connects two coupled sections. Like edge coupled filters, the resonators are 180 degrees long, and as a big advantage, no grounding is needed.

Hairpin resonators are U-shaped and they alternate up and down for orientation. Hairpin filter is based on the coupling between U-resonators. The undesired coupling between arms of U can pose a problem if not properly controlled. In iFilter, the distance between arms is set to 2*W, which limits the coupling to less than 25dB.

Hairpin filters are only built on microstrip and stripline. For other media, supporting U-resonators is not a feasible idea.

Typical Specifications

- **Approximation**: Chebyshev only
- **Degree**: 3 to 8
- **BW**: up to 40% (requires more tuning and tight line spacings as BW% gets higher)
- **Resonator Zo**: 40 to 90 ohms (must be realizable)

Tuning and Optimization

- LTOT - is the length of one arm of the U-shaped hairpin resonators. Increasing/decreasing it moves the center frequency of the filter down/up respectively.
- LBot - is the tapping length at the input and output measured from the bend of the U towards the open end. Increasing or decreasing LBot causes the passband return loss to go up and down. For speed purposes, the tapping length cannot be calculated very accurately. You should tune this parameter in the Microwave Office program in a few seconds. Other lengths are formulated to keep the center frequency constant as much as possible while tuning LBot.
- L_v3 (usually) - is the outermost line extension. It represents the amount of line length to add or deduct from the outermost U-section arms. This parameter is usually negative, meaning that the outermost arms should be a little shorter than the inner lines. Decreasing this parameter (making it more negative) usually reduces the upper corner end return loss to even out across the passband. There is no explicit formulation available for this parameter. In fact, examples in the literature assume that it is a positive length. The best approach is to tune it in the Microwave Office program which takes just a few seconds.
- S1_v1 - is the spacing between arms of U-shaped resonators. It is conventionally taken as twice the resonator width. Tuning this parameter does not affect the response much for most designs. When the dielectric constant is quite high, some adjustment is expected.
- S2_v1 - is the spacing between the first and second U-resonators (from both ends mostly). Tuning this affects the passband return loss. Passband width is usually not affected by S2_v1.
- All other spacings - as Sx_v1 are the inner resonator spacings. Increasing them widens the passband and decreasing them narrows it. To achieve a well-matched response for the desired bandwidth, you should tune all inner spacings at the same time. Decreasing one spacing distorts the return loss while widening the passband. Decreasing another spacing widens the passband more, but also decreases the return loss to the desired level.

13.6.9. Bandstop Filters

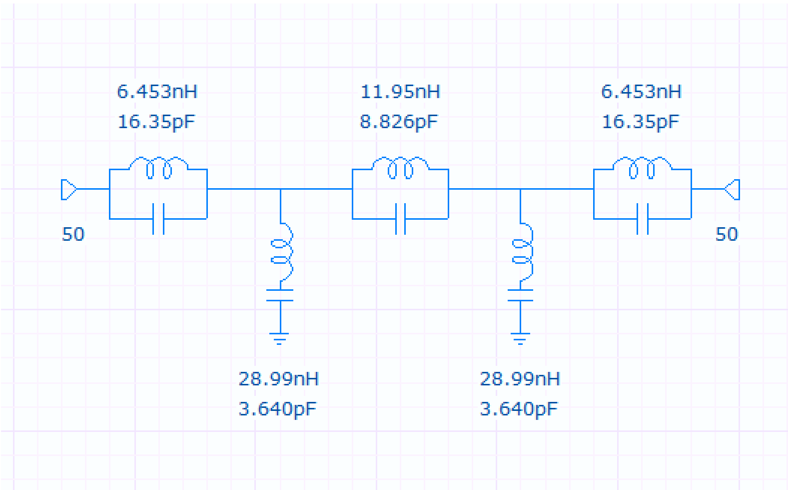
Bandstop filters are designed to stop signals within a specific frequency range and pass all signals outside of that frequency range. For simplicity, bandpass filter terminology is applied to bandstop filters. The center of the stopband is called Fo and the stopband width is called BW. Here, BW is measured from lower passband cutoff frequency to upper passband cutoff frequency. For filters that have equi-ripple passband characteristics (Chebyshev, Elliptic, and Generalized Chebyshev), this corresponds to the passband ripple corner. For other approximation types it is the 3.011 dB (3-dB) corner frequency.

Bandstop filters are designed to provide infinite attenuation in the stopband center. The depth which the bandstop filter can achieve depends on the quality. Just as bandpass filters have insertion loss due to lossy elements and parasitics, bandstop filters cannot provide infinite or zero impedance, so there is always a leakage towards load side. To achieve very narrowband bandstop filters, you needs high Q elements, just as for very narrowband bandpass filters.

For distributed element filters, the response repeats itself due to the periodicity of electrical lengths, so you can also use a distributed lowpass filter as a bandstop filter.

13.6.9.1. Lumped Element Bandstop Filter

Lumped element bandstop filters contains series PLC and shunt SLC elements. For filters with finite TZs, SLC resonators are replaced by quad LC-resonators, which are simplified to 2 shunt SLC resonators. A Chebyshev bandstop filter is shown in the following figure.



Realization of lumped element filters is defined in "Lumped Model Options Dialog Box ".

Typical Specifications

- **Approximation**: All available
- **Degree**: See standard ranges

13.6.9.2. Optimum Distributed Bandstop Filter

Distributed element bandstop filters are identical to optimum distributed lowpass filters with the specification reformatted. A bandstop filter is defined by center frequency Fo and bandwidth BW. An optimum lowpass filter is defined with EL at passband corner Fp. At 90 degrees long, the lowpass filter has no transmission. The S21 response is re-entrant at 90+EL as the corner of spurious passband. The following relations can therefore be established between two filters:

Fo = Fp * 90/EL

BW = 2*(Fo-Fp)

Optimum distributed bandstop filters are not suitable for narrow bandwidths. The shunt open-circuited stub impedances come out very high for bandwidths below 40-45%. Therefore, this filter is more suitable for wide bandwidths.

Typical Specifications

- **Approximation**: Chebyshev only with fixed 26dB, 20dB or 16dB return loss.
- **Degree**: 3 to 19
- **BW**: 10% to 160%

Tuning and Optimization

- L_v(n) - are the element line lengths. Slight tuning may be required for the lengths for accurate passband corner and return loss.

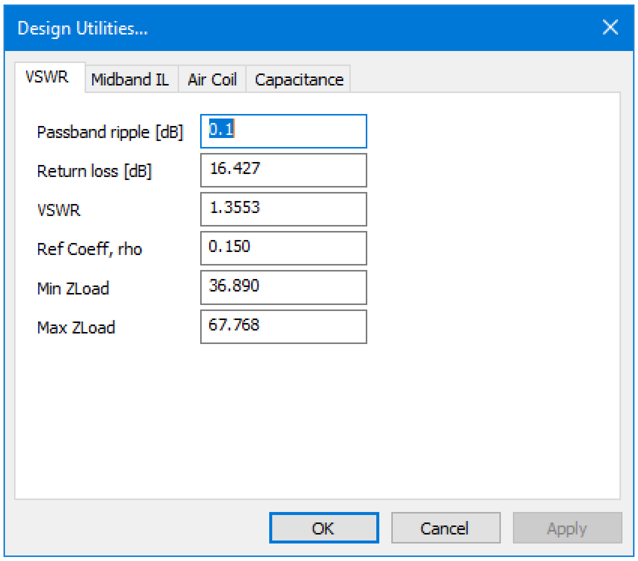
13.6.10. Auxiliary Dialog Boxes

iFilter includes settings and utility dialog boxes that you can display and use any time during the filter design process.

13.6.10.1. Design Utilities Dialog Box

The Design Utilities dialog box contains basic filter-related conversion and calculation utilities. To display this dialog box, click the **Design Utilities** button in the main iFilter dialog box.

Design Utilities VSWR (Conversion) Tab



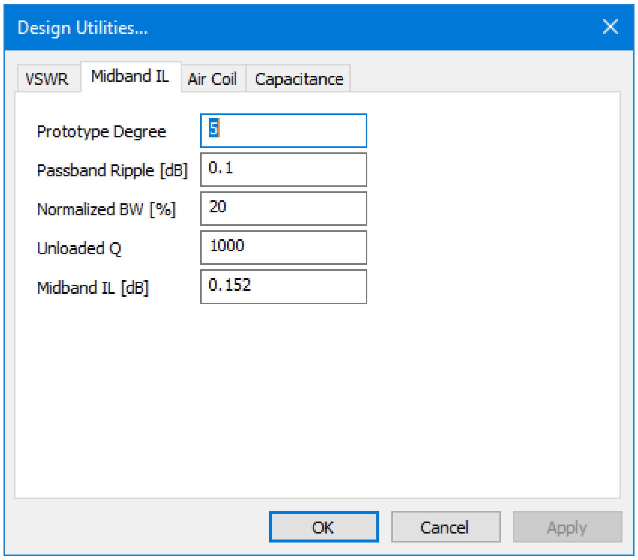
This dialog box tab converts values between well-known filter parameters. To view this tab, click the **Design Utilities** button in the main iFilter dialog box and then click the **VSWR** tab. The following parameters are available for converting:

- **Passband ripple**
- **Return loss**
- **VSWR**
- **Ref Coeff** (Reflection coefficient)
- **Min ZLoad** and **Max ZLoad** (Minimum and Maximum Load Termination)

Entering any of the first three parameters concurrently updates all other values in the dialog box.

NOTE: The passband ripple is associated with the filter's insertion loss. It is named so as not to be confused with the dissipated loss. You can use the value for any insertion loss caused by impedance mismatch.

Design Utilities Midband IL (Midband Insertion Loss) Tab



This dialog box tab estimates the midband insertion loss of a bandpass filter. To view this tab, click the **Design Utilities** button in the main iFilter dialog box and then click the **Midband IL** tab.

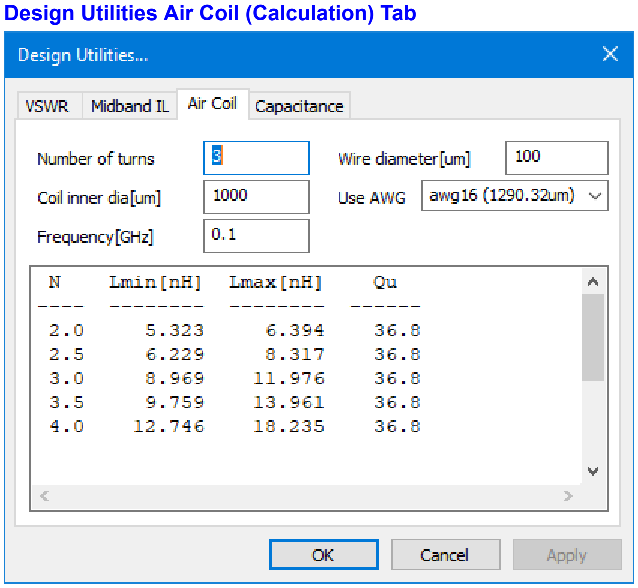
The insertion loss of a bandpass filter can be approximated by the formula:

$$IL = \frac{4.343 \sum g_k}{Q_u * bw}$$

This formula associates the unloaded Qu of filter elements and the filter Q (Fo/BW) to the loss. Although initially developed for narrowband microwave filters, the concept is applicable to all bandpass filters made up of any technology.

When you edit any of the parameters, the dialog box concurrently calculates and updates the midband insertion loss. If you edit the midband IL, the required unloaded Q is calculated and updated.

Design Utilities Air Coil (Calculation) Tab

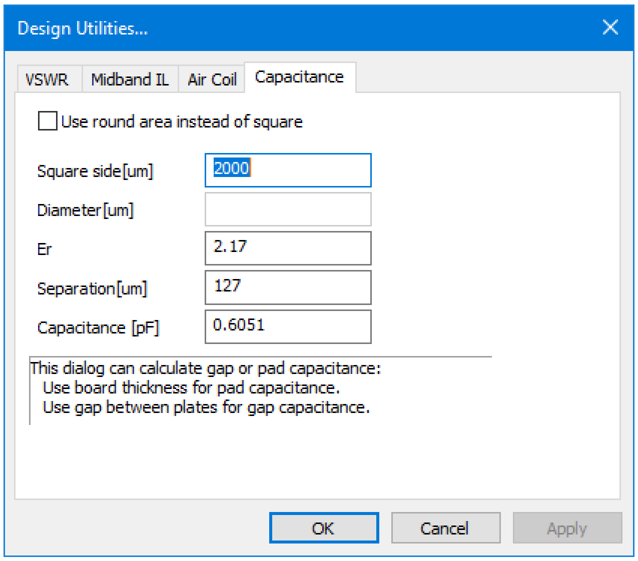


This dialog box tab helps you calculate the parameters of an air coil inductor. To view this tab, click the **Design Utilities** button in the main iFilter dialog box and then click the **Air Coil** tab.

For lumped element filters, you may need to use wound air coils. On this tab, you can conveniently calculate inductance and Q values. The minimum and maximum coil inductances are based on the gap between the turns. Lmax is given for 0.5wd (tight winding) and Lmin is given for 2wd (loose winding) where the variable `wd` defines the wire diameter. Although the formula is very accurate, the actual inductor value depends on the orientation and proximity of the coil to the housing.

Qu (unloaded Q) is more empirical, but gives you an idea of the order of magnitude for a realizable Qu.

Design Utilities Capacitance (Gap/Pad) Tab



Two parallel conducting plates separated by a distance *d* create a static capacitance:

$$C[pF] = \frac{8.854\varepsilon_r A}{d}$$

where *A* is the area of the plates, and *Er* is the dielectric constant of the gap medium. Both *A* and *d* are in meters.

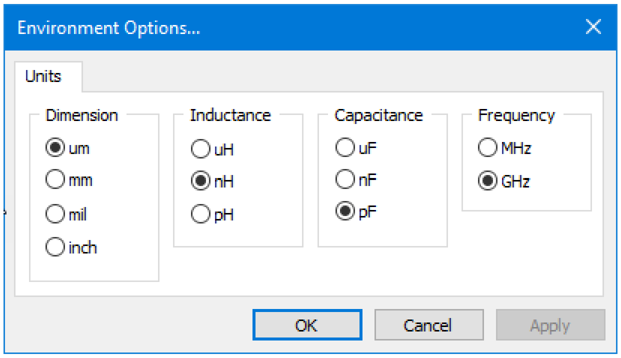
The separation may be air or a dielectric. Lumped element filters are built on printed circuit boards. Pads are needed on PCB where the coils should be connected to capacitors. Pads have a certain area where they create a capacitance to the bottom of the PCB. At high frequencies, capacitors of the filter elements reduce to few pF and below. In these circumstances, capacitance due to pads becomes comparable and may increase or decrease the effective capacitance of the filter elements.

For example, a 4mm2 pad on a 0.005” Duroid has 0.6pF. This should be included in a detailed design as if there is shunt capacitance to ground from every pad of 0.6pF.

13.6.10.2. Environment Options Dialog Box

The Environment Options dialog box displays unit settings and general settings. To view this dialog box, click the **Environment Options** button in the main iFilter dialog box.

Environment Options Units Tab



The iFilter units settings are initiated from the Microwave Office program, however when iFilter is running, you can set the units locally to the wizard. The settings are saved in each design separately, so if you load a previous design, the displayed units change to those of that design.

13.6.11. Design Examples

To demonstrate iFilter capabilities and simplify the wizard's functionality, the following examples are included.

13.6.11.1. Lumped Element BPF Example

This is a lumped bandpass filter design. Without targeting a specific application, the specification is defined as:

- 15 dB minimum return loss in a passband of 475 to 525 MHz
- Maximum insertion loss of 4 dB
- 50dB attenuation at 600 MHz

1. Open the **Wizards** node in the Project Browser and double-click the **iFilter Filter Synthesis**. The main iFilter dialog box displays with properties from any previous design.
2. Click the **Change Filter Type** button at the top left of the dialog box in the **Type-Approximation** area. The Select Filter Type dialog box displays.
3. Click the **Bandpass** button and then click the **Lumped** button in the row beneath it.
4. In the **Main Filter Type** list, select **Narrowband Lumped Filter**. In the **Options** list, select **Capacitive (identical shunt L)**, then click **OK** to close the dialog box.

   You now return to the main iFilter dialog box. The filter type changes but the common properties of the previous design still display.
5. If the **Change Response Approximation Type** button (second button from the top) does not display "Chebyshev", click the button to display the Approximation Function dialog box, select **Chebyshev** and close the dialog box.
6. In the **Ripple** text box, type "**0.01**", which corresponds to 16.4dB return loss.

   For **Degree** type "**5**".

   For **Fo** type "**500**".

   For **BW** type "**50**".

   You may see a plotted response.

7. Click the **Environment Options** button in the main iFilter dialog box to display the Environment Options dialog box. Click the **Units** tab and ensure that **Frequency** is set to **MHz** and **Dimension** is set to **mm**.
8. Click the **WS** (Wide Analysis Span) button in the row of buttons under the plot to set the wideband span to the chart.
9. To see actual marker values, click the **Edit Chart Settings** button at the top left of the plot. In the Chart Settings dialog box, click the **IL+RL** button, then click the **Markers** button.
10. Click the **Add** button to display the Add Marker dialog box. For **Fmin** type "**500**", then select **Insertion Loss**. Repeat this step for an **Fmin** value of 600 MHz, and then click **OK** to close the dialog box.

    The markers now display on the insertion loss trace.

    Unfortunately, the filter does not have the required 50dB attenuation at 600 MHz; it has only 45dB of attenuation. By increasing **Degree** to 6, you can get 61dB attenuation at 600MHz. Increasing the filter order adds extra elements to the circuit, but gives you more attenuation than needed, and also increases the midband insertion loss. By widening the bandwidth of the filter, you can find a suitable attenuation and less insertion loss. Increasing **BW** to 56 MHz, you now have 55dB attenuation with 5 dB cushion above the specification.

    The filter has shunt resonators coupled with capacitors. The resonators have identical shunt inductors, which is ideal as you only need one inductor value for the filter design. An air wound coil of 1 to 1.5 turns of AWG #32 wire with 2mm coil diameter provides about 2 nH of inductance. The same dialog box reports QL=164 at 500 MHz. A commercially available capacitor has QC=350 at 500 MHz. By setting these values to QL and QC respectively, and clicking the **Analyze Lossy** button, you can approximate the response with more realistic nonideal conditions. The lossy simulation now shows 3.6dB insertion loss with 55dB attenuation.
11. Click the **Design Options** button to set element realization options. In the Lumped Model Options dialog box, select **Use AIR COIL** and **Coil (1-10nH)** for the inductor value range of 1-10nH. Set **Rmax** to "**2**" mm for coil diameter and **Wire** to **awg32**.
12. Select **Vendor Part** and then under **Capacitors** select **Vendor 0-10pF** and **Vendor 10-100pF** for capacitor value ranges 0-10pF and 10-100pF. Click the **Capacitor Vendors** button to display the Capacitor Vendors and Parts dialog box. In this dialog box, select your preferred vendors along with the part sizes. For 500MHz, Cadence recommends sizes up to and including 0805.
13. Click the **Search for available types** button and select 600L and 600S series and move them to the **Selected Types** list by clicking the right arrow. Click **OK** to close the dialog box, then close the Lumped Model Options dialog box as well.

    iFilter searches through the selected part types and finds the most suitable parts with highest Q.
14. Click the **View Circuit Information** button to see what components are selected for the current schematic. This view details the recommended parts together with the variation of element values. To see the effects of the real-life components, click the **Analyze Real** button in the Analysis group. The response changes slightly from the lossy model. It may shift along frequency axis and may scale up or down depending on the available Q. If the response becomes significantly different than ideal or lossy models, it is most likely due to the insufficient modeling of the vendor parts selected.
15. Click the **Generate Design** button to generate the design in the Microwave Office program. In the AWR Design Environment platform you can perform optimization and yield analysis. A PCB layout can be drawn up and exported CAD/CAM data can be generated for the board manufacturer.

13.6.11.2. Microstrip Bandpass Filter Example

This is a tapped input edge coupled filter design. The specification is defined as:

- 15 dB minimum return loss in a passband of 4500 to 5500 MHz

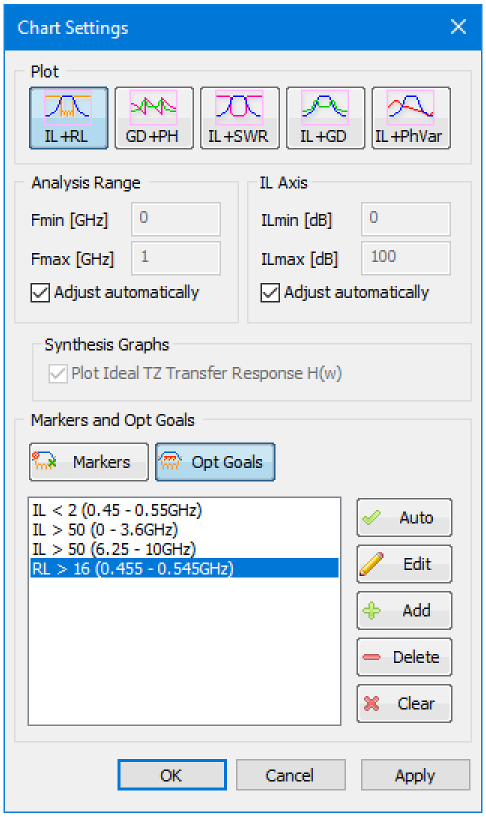
- Maximum insertion loss of 2 dB
- 50dB attenuation at 3600 MHz

1. Open the **Wizards** node in the Project Browser and double-click the **iFilter Filter Synthesis**. The main iFilter dialog box displays with properties from any previous design.
2. Click the **Change Filter Type** button at the top left of the dialog box in the **Type-Approximation** area. The Select Filter Type dialog box displays.
3. Click the **Bandpass** button and then click the **Microstrip** button in the row beneath it.
4. In the **Main Filter Type** list, select **Edge Coupled Bandpass Filter**. In the **Options** list, select **Tapped input/output**, then click **OK** to close the dialog box.

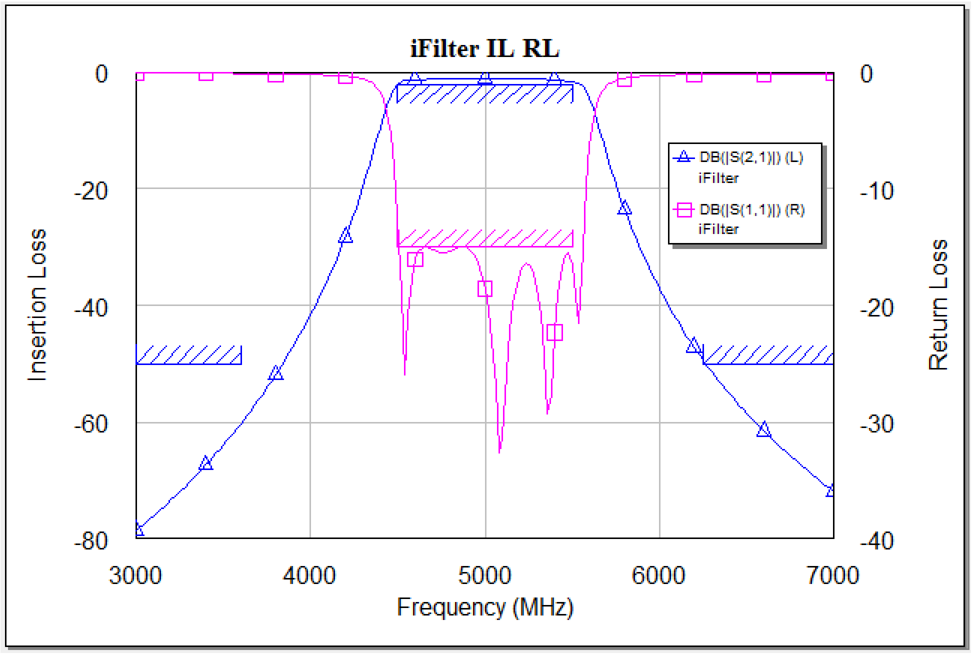
   You now return to the main iFilter dialog box. The filter type changes but the common properties of the previous design still display.
5. The Tapped input/output type is optimized for 20dB return loss, so click the text box to the right of the **Ripple [dB]** button and scroll the mouse wheel until the value is 0.0436dB. A 6th order filter with Fo=5000MHz and BW=1000MHz meets the required specs.

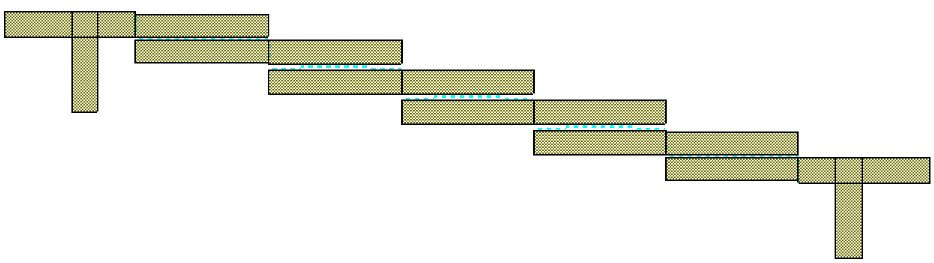
   Next is the selection of a suitable substrate. A major selection criterion is typically the cost of the material. Some substrates are cheaper to purchase and process, but these may perform well. Electrically, the quality factor of the substrate that affects the insertion loss is the most important property to be aware of. The quality factor is determined by two losses: Conductor loss and Dielectric loss. Conductor loss is a function of skin depth and line width. As line widths are proportional to the substrate height, selecting thicker substrates yield wider line widths, so low insertion loss. Dielectric loss is substrate-dependent and is determined by the **TanD** (dielectric loss tangent) parameter. In iFilter, you can specify the **TanD** parameter, however it is not taken into account for determining the dimensions. When the final simulation is performed using the Microwave Office program to calculate the filter response accurately, **TanD** is used. Naturally, a low **TanD** yields better insertion loss in the passband.
6. In this design example, assume a RO4350B commercial material is available with Er=3.48, 0.030-inch thickness (0.762mm). To enter the parameters, click the **Design Options** button and click the **Technology** tab in the Distributed Model Options dialog box. Enter the substrate **Height**, click in the **Substrate Er** text box, and scroll the mouse wheel until the 3.48 displays. **TanD** displays a value of 0.0037. Select a suitable **Cond.Thickness** (conductor thickness). A ½ oz. copper, corresponds to 17um (0.017mm or 0.67mil).
7. In the same dialog box, click the **Realization** tab and select **Add input and output lines to the layout** to add reference 50 ohms lines to both sides of the filter. Clear the other check boxes and click **OK**. The filter is ready to analyze and optimize.

8. Click the **Edit Chart Settings** button at the top left of the plot. In the Chart Settings dialog box, click the **OptGoals** button, then click the **Auto** button to automatically add some OptGoals to the design. Select the OptGoal from list which determines lower stopband with 50dB spec. Because iFilter calculates 50dB points from ideal attenuation functions, you should edit this. You should also edit Return loss to cover 15dB only, rather than the designed 20dB.



9. Click the **Generate Design** button and export the filter into the Microwave Office program with response. iFilter generates a design with schematic, graph(s) and optimization goals already set. Due to accurate analysis of microstrip TEEs and steps, the return loss is slightly worse than that shown in iFilter.
10. In the AWR Design Environment platform, choose **Simulate > Optimize**. In the Optimizer dialog box, check only **L_v2**, **L_v3** and **S_v1**, the most effective parameters on the return loss. Click the **Start** button to quickly converge to a solution.





11. You can now try the effect of **TanD** on the insertion loss. At 0.0037, IL is 1.01dB. Increasing **TanD** to 0.01 increases the insertion loss 2.05dB. This demonstrates the importance the dielectric loss tangent is at high frequencies.

13.6.11.3. Arbitrary Narrowband Filter Simulation Example

You can simulate narrowband microwave bandpass filters with lumped elements if you use the proper coupling and enter unloaded Q values correctly.

A combline filter is quasi-lowpass in nature (inductively coupled). Its selectivity in the upper stopband is better than that of the lower stopband. If you select an inverter-based filter type with inductive coupling, you get series inductors and shunt LC-resonators in your filter. Although a combline filter is made in an entirely different medium with metal bars and screws etc., the lumped element filter's stopband response closely resembles the combline filter.

You should set the LC-resonator element Q's slightly higher than what can be obtained with a combline structure to give the passband response of the hypothetical combline filter. To do so, click the **Design Options** button and click the **Realization** tab. Select the **For BPF: set loss factors to shunt resonators only** check box, then enter approximately 1.6 times higher Q (obtained for combline structure) for both QL and QC. The number 1.6 is obtained after some sample designs, but can also be changed to your previous findings.

The response should now provide an idea of the combline filter's response, both in passband and stopband. By placing markers, you may obtain insertion loss values at any frequency.

This method is valid for any microwave filter with known coupling type and unloaded Q-value. Types of microwave filters range from low-Q LTCC's to dielectric resonator with very high element Q's. This method is more advanced, yet equally simple as the industry standard midband loss method, which only addresses the passband center.

13.7. iFilter Synthesis Wizard

iFilter Synthesis is a filter synthesis capability embedded within the Cadence® iFilter™ filter synthesis wizard that allows designers to:

- place transmission zeros to form a transfer function H(s)
- extract filter elements by removing poles from H(s)
- perform network transformations for easier realization

Standard iFilter uses known topologies for its filter solutions with explicit formulations where element values are often realizable. Many filters used in RF and microwaves are in the form of a fixed topology, so iFilter is suitable for most applications.

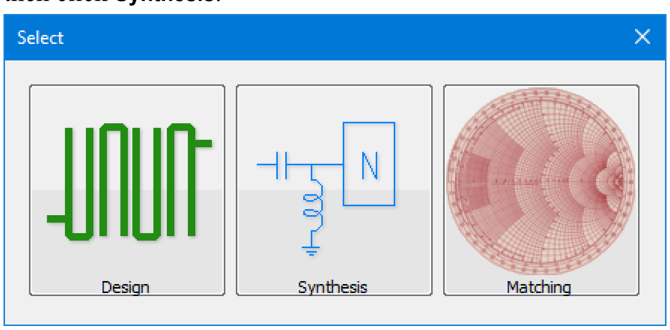
iFilter, however, does not have built-in solutions in design constraints such as the following, when:

- the specifications require a specific selectivity slope or full reflection (S21 = 0) at some frequencies
- the topology is not suitable (for instance, series short-circuited stubs on microstrip)
- the topology has wide-ranging element values (like inductor values changing from uH to pH in the same filter)

In this case, the iFilter Synthesis presents various options and offers considerable flexibility.

13.7.1. Running the iFilter Synthesis Wizard

To run the iFilter Synthesis Wizard, open the **Wizards** node in the Project Browser and double-click **iFilter Filter Synthesis**, then click **Synthesis**.

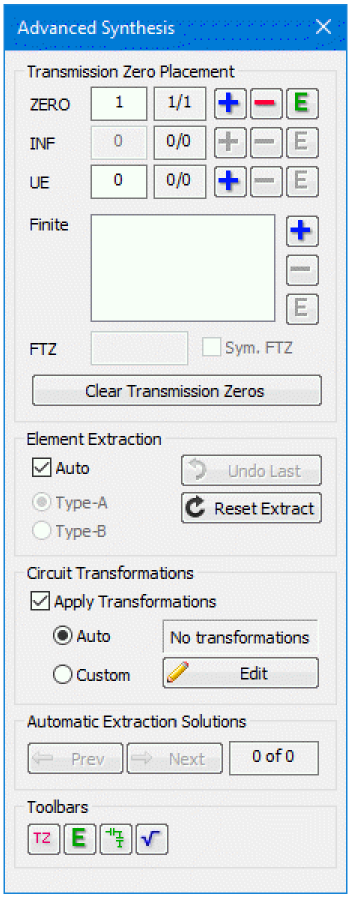


13.7.2. Synthesis Specific Dialog Boxes

The following dialog boxes and toolbars are used in filter synthesis.

13.7.2.1. Advanced Synthesis Dialog Box

The Advanced Synthesis dialog box is used in manual or semi-automatic synthesis design mode.



This dialog box is laid out vertically in the order of synthesis steps with the following sections:

- Transmission Zero Placement - manual
- Element Extraction - manual/automatic
- Circuit Transformations - manual/automatic
- Automatic Extraction Solutions - automatic

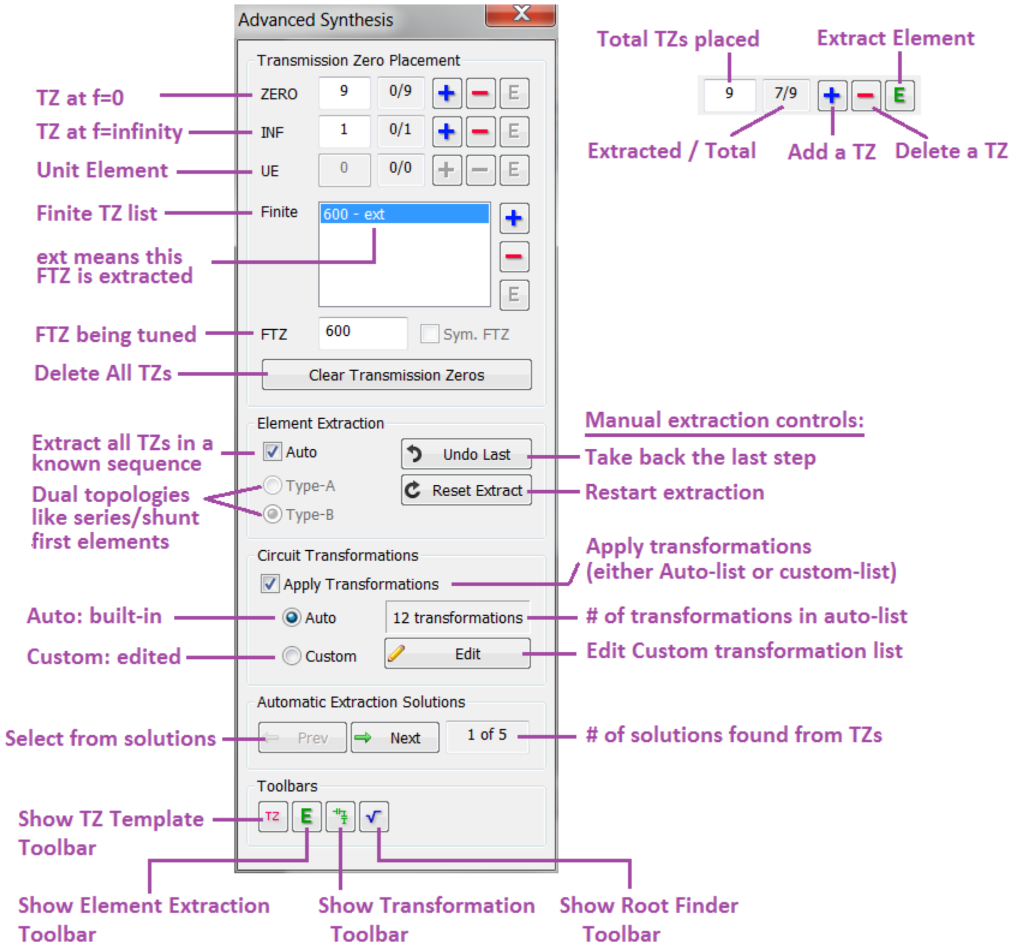
The "solutions" are a combination of the extraction sequence and circuit transformations. So, for a given list of transmission zeros, there can be several ways to extract the elements, and a suitable set of transformations for each extraction sequence. If there is an applicable solution, they are listed at the bottom of the dialog box, for example "1 of 5" shown in the figure.

Automatic actions are triggered as a result of two conditions:

- when an appropriate control is selected
- when the specification or topology is suitable to perform the action

In the previous figure, solution 1 is selected out of 5 potential solutions. Since the **Auto** check box is selected, the elements are extracted using the pre-stored extraction sequence for this solution. Since the **Apply Transformations** check box is also selected, the transformations given in Solution #1 are also applied to the filter.

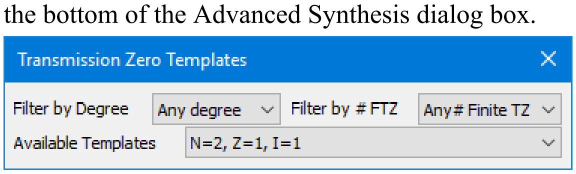
The annotated Advanced Synthesis dialog box is shown in the following figure.



The toolbar buttons at the bottom of the dialog box are provided for convenience and are optional.

13.7.2.2. Transmission Zero Templates Toolbar

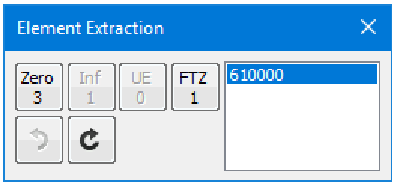
iFilter Synthesis has pre-stored solution templates for various sets of transmission zeros. The Transmission Zero Templates toolbar provides easy access to those templates. To display this toolbar, click the **Show TZ Templates Toolbar** button at the bottom of the Advanced Synthesis dialog box.



The available templates are listed in the drop-down list. If you select a new item, the corresponding TZs are displayed in the Advanced Synthesis dialog box. The templates can be filtered by degree or number of finite TZ.

13.7.2.3. Element Extraction Toolbar

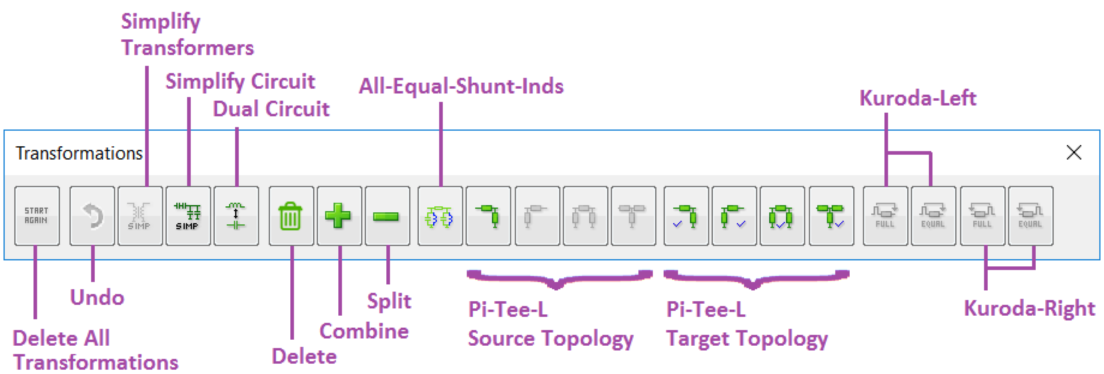
The Element Extraction toolbar is another way of extracting circuit elements from the synthesis function. To display this toolbar, click the **Show Extraction Toolbar** button at the bottom of the Advanced Synthesis dialog box.



The three left buttons in the top row are used to extract TZ from f=0, f=infinity and a Unit element. The fourth button is used to extract the selected FTZ from the list. These buttons replicate the buttons in the Advanced Synthesis dialog box. The left button in the bottom row undoes the last transformations and replicates the **Undo Last** button in the Advanced Synthesis dialog and the **Reset Element Extraction** button replicates the **Reset Extraction** button.

13.7.2.4. Transformations Toolbar

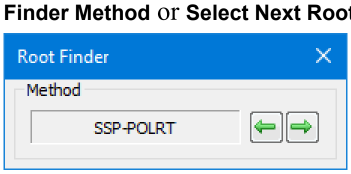
The Transformations toolbar provides shortcuts to the common transformations that are normally listed in the Circuit Transformations dialog box. To display this toolbar, click the **Show Transformation Toolbar** button at the bottom of the Advanced Synthesis dialog box.



Most of the transformations in the toolbar work immediately. Pi-Tee-L transformations are slightly different. Since there are many different combinations of these transformations, and not all of them can be displayed, only the symmetric impedances cases are shown. In order to apply Pi-Tee-L transformations, you must first select the source topology and then the target topology. For example, to apply “LLeft to Pi – Symmetric Imp”, first select the LLeft from the source topology group, and then select Pi from the target topology group.

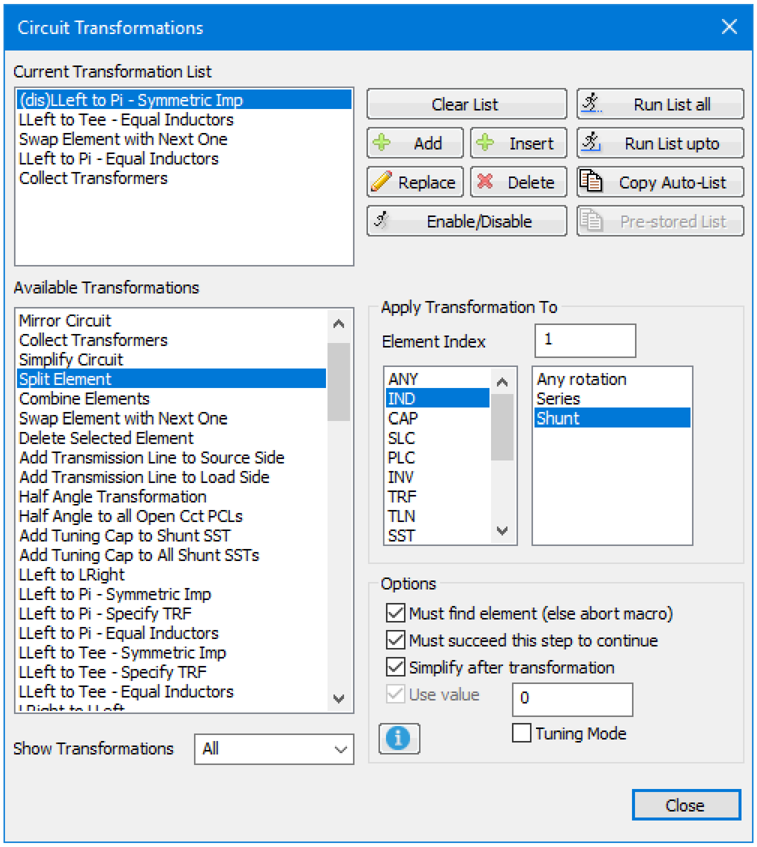
13.7.2.5. Root Finder Toolbar

The filter synthesis process requires many math operations, including complex number root finding. There is no single root finding algorithm that works for all combinations of transmission zeros. iFilter Synthesis provides several root finding algorithms to choose: SSP-POLRT, Newton, Bairstow, and Jenkins. SSP-POLRT is suitable for most of the cases so it is the default algorithm. You can change the selected algorithm by clicking the arrow buttons (**Select Prev. Root Finder Method** or **Select Next Root Finder Method**) on the Root Finder toolbar.



13.7.2.6. Circuit Transformations Dialog Box

To access the transformations of the current filter, click the **Edit** button in the Advanced Synthesis dialog box to display the Circuit Transformations dialog box.



The **Current Transformation List** displays the transformations applied to the extracted filter circuit. If the filter is in Auto-Transformation mode (**Auto** is selected in the Advanced Synthesis dialog box), the transformations display but are not editable. To allow editing in Custom-Transformation mode, select **Custom** in the Advanced Synthesis dialog box.

This dialog box includes the following options:

- **Clear List** deletes all the transformations in the **Current Transformation List**.
- **Add** adds the selected transformation in the **Available Transformations** list to the **Current Transformation List**.
- **Insert** inserts the selected transformation in the **Available Transformations** list to the **Current Transformation List** above the selected item.
- **Replace** replaces the selected transformation in the **Current Transformation List** with the selection in the **Available Transformations** list.
- **Delete** deletes the selected transformation in the **Current Transformation List**.
- **Enable/Disable** toggles the selected transformation in the **Current Transformation List** as enabled or disabled without deleting it.
- **Run List all** runs all the transformations in the **Current Transformation List**.
- **Run List upto** runs transformations up to the selected one in the **Current Transformation List**.

- **Copy Auto-List** copies the transformations from the pre-stored Auto list into the **Current Transformation List**.
- **Available Transformations** lists all the available transformations in the iFilter library. When there are too many to choose from, you can display a subset of transformations by selecting the desired option in **Show Transformations**.

The **Options** section of the dialog box lists options that are complementary to the selected transformation. In the previous figure, the setting for applying the transformation to the “1st shunt capacitor” is shown.

The software applies the transformations in the order shown in the **Current Transformation List**. If the software cannot find the element to apply the transformation, it may either abort the whole list or continue. To stop if a required element is not found, select **Must find element (else abort macro)**.

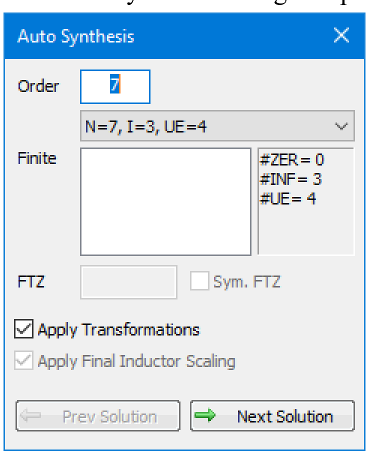
If you want the software to stop upon a transformation fail because the following transformations depend on its success, select **Must succeed this step to continue**.

Some transformations result in an extra transformer. To simplify automatically after the transformation, select **Simplify after transformation**. This option is equivalent to adding an extra Collect Transformers command to the **Current Transformation List**.

Some transformations require you to enter a value with the command. For example, for Split Element you need to specify one value. To specify a value, select **Use value** and enter a value in the text box. If **Tuning Mode** is selected, you can change the value while the full transformation list is continuously applied to the filter.

13.7.2.7. Auto Synthesis Dialog Box

The Auto Synthesis dialog box presents simple controls for synthesizing in the automatic mode.



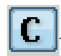
This dialog box includes the following options:

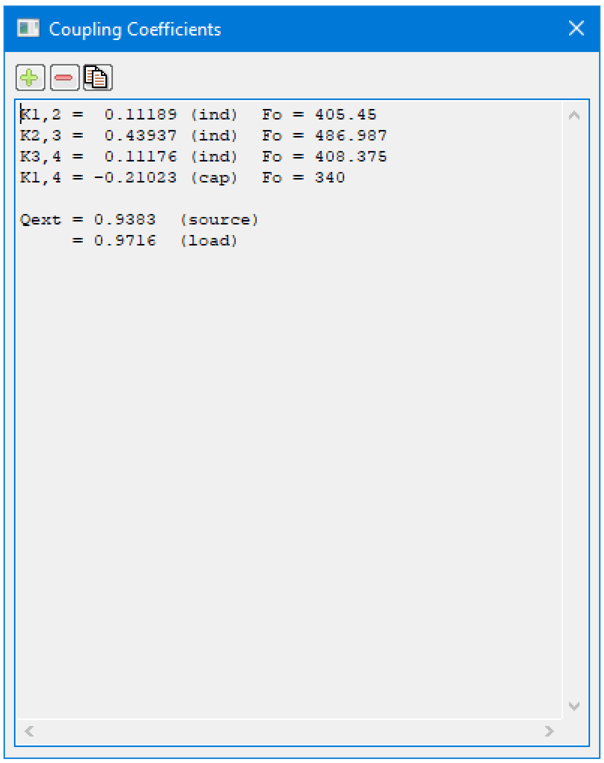
- **Order** specifies the filter order.
- The drop-down list below **Order** lists templates that change depending on the order specified. In the previous figure, N=6, Z=3, I=1, F=1 (0 ls, 1us) indicates the 6th order filter with 3 TZ at f=0, 1 TZ at f=infinity, and 1 finite TZ (1 at upper stopband). If the drop-down list changes the wizard resets the TZ list and creates finite TZs in the right side of the passband. In this example, 1us places the single FTZ at 610MHz, which is above upper frequency corner of the bandpass filter.
- **Finite** lists the finite TZs. You can tune the FTZ selected in the list higher and lower in frequency.
- **Sym. FTZ** is rarely used. The most appropriate case is when designing topologies with CQ-sections which require FTZ pairs which are symmetric around filter passband.

- **Apply Transformations** is selected for all automatic synthesis. It is provided as a test option to check on the raw extraction circuit process.
- **Prev Solution** and **Next Solution** buttons are used to select from pre-stored solutions that are suitable for the selected TZ template.

13.7.2.8. Coupling Coefficients

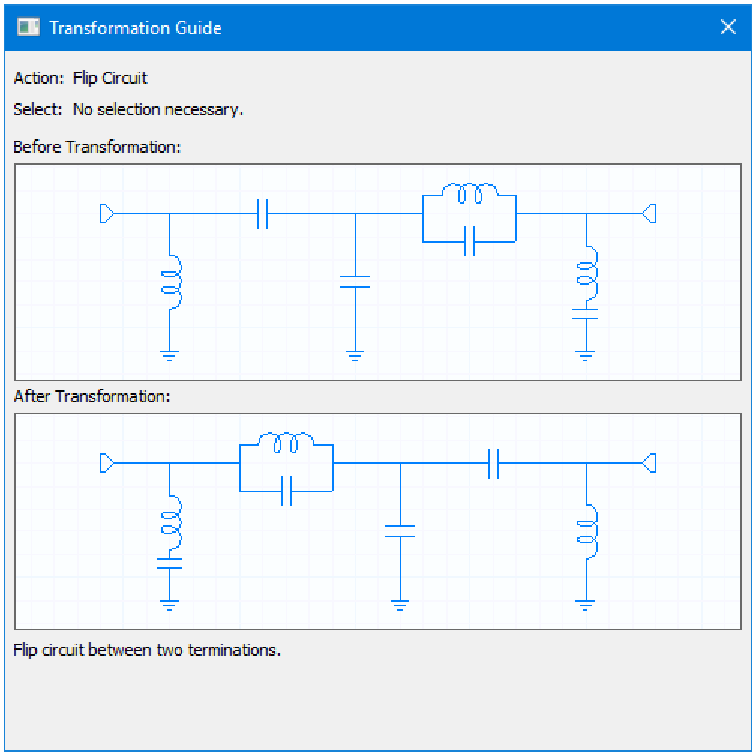
Most narrowband bandpass filters are realized using cavity combline resonators. These filters are normally realized by first selecting a suitable resonator impedance and then placing resonators by coupling them through irises. The relation between the iris dimensions and coupling coefficients are established through measurements. The coupling coefficients are found from the equivalent filter circuit. iFilter Synthesis displays coupling coefficients in a separate window.

To display the Coupling Coefficients window, click the button in the main iFilter dialog box.



13.7.2.9. Transformation Guide Dialog Box

The Transformation Guide dialog box provides quick information for transformations. To display this dialog box, click the button in the Circuit Transformations dialog box.

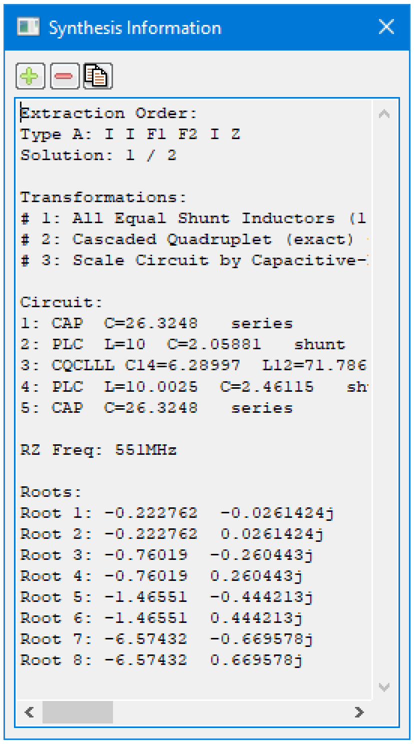


While this dialog box displays, you can select any transformation from the **Available Transformations** list in the Circuit Transformations dialog box to view its information. The schematic in the **Before Transformation** pane shows a typical topology for the transformation to be applied. The **After Transformation** pane shows the schematic after transformation. If the transformation requires you to select an element to apply, the element is marked in the schematic. To apply the transformation, you must select that element in the filter.

13.7.2.10. Synthesis Information Window

iFilter Synthesis provides a summary of synthesis actions in an information window. To display this window, click the 

button in the main iFilter dialog box.



13.7.3. Lumped Bandpass Filter Example

The following examples explain iFilter Synthesis functionality. Here, you design a 5th-order bandpass filter, centered at 500MHz, with a 40MHz bandwidth and a Chebyshev response with a 0.01dB passband ripple.

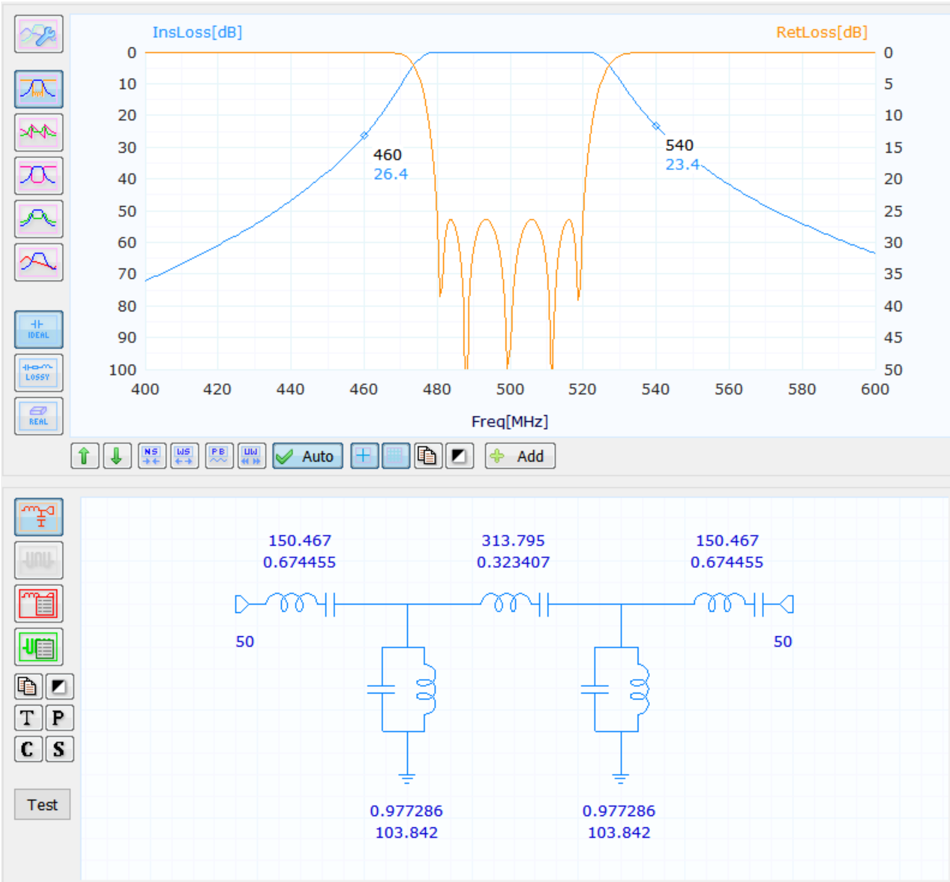
13.7.3.1. Solution 1 - Standard Textbook Solution from iFilter

A well-known passive filter design technique starts by constructing a lowpass prototype that is normalized to 1 ohm terminations and a 1Hz cut-off frequency. Next, a frequency transformation is applied for the required passband, and finally an impedance scaling is applied to the circuit elements, and all impedances are multiplied by the value of source impedance, which is usually 50 or 75 ohms in RF and microwave systems.

A transmission zero (TZ) is defined as a frequency where there is no transmission (that is, the input signal is fully reflected,$|S11|=1$ and $|S21|=0$. An Nth order lowpass prototype contains N transmission zeros at $f=\infty$. When a frequency transformation is performed on the lowpass prototype, transmission zeros are also moved to new frequencies. An Nth order lowpass prototype results in the following after transformation:

- For lowpass: There are N TZs $f=\infty$
- For highpass: There are N TZs at $f=0$
- For bandpass: There are N TZs at $f=0$ and N TZs at $f=\infty$ because prototype TZs are mapped to both sides of the passband. Therefore, bandpass filters have 2N transmission zeros shared between upper and lower stopbands.

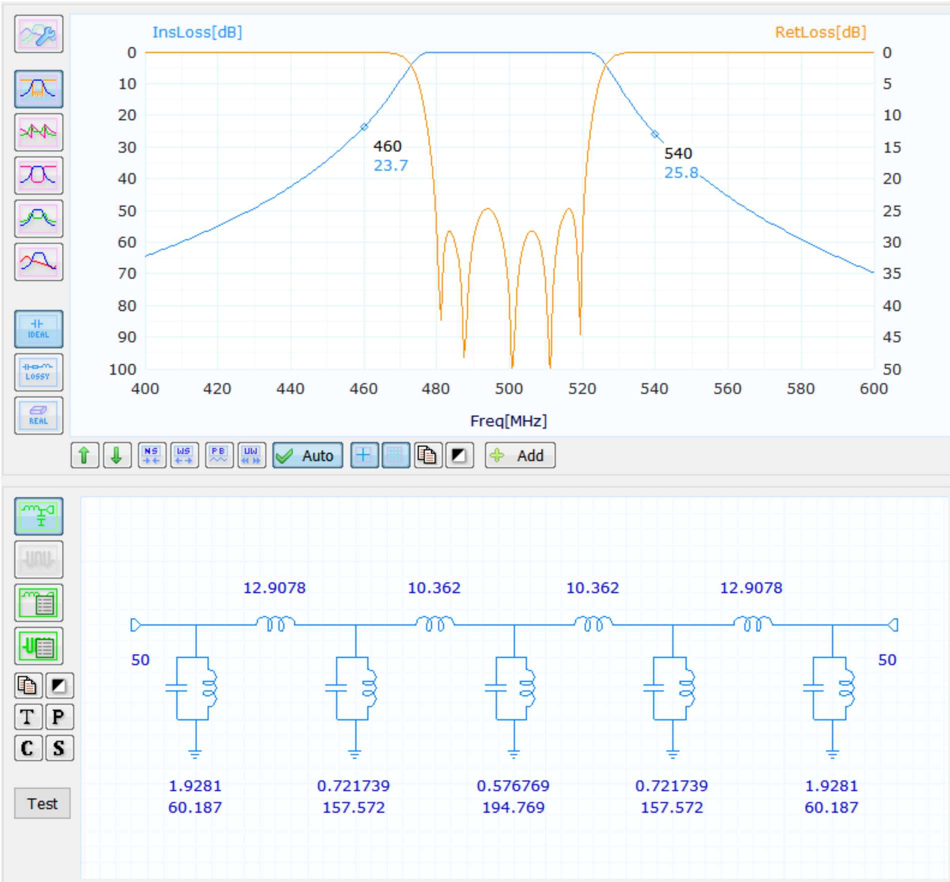
The following figure shows a standard textbook bandpass filter obtained by selecting Bandpass > Lumped > Lumped Element Filter in the Select Filter Type dialog box.



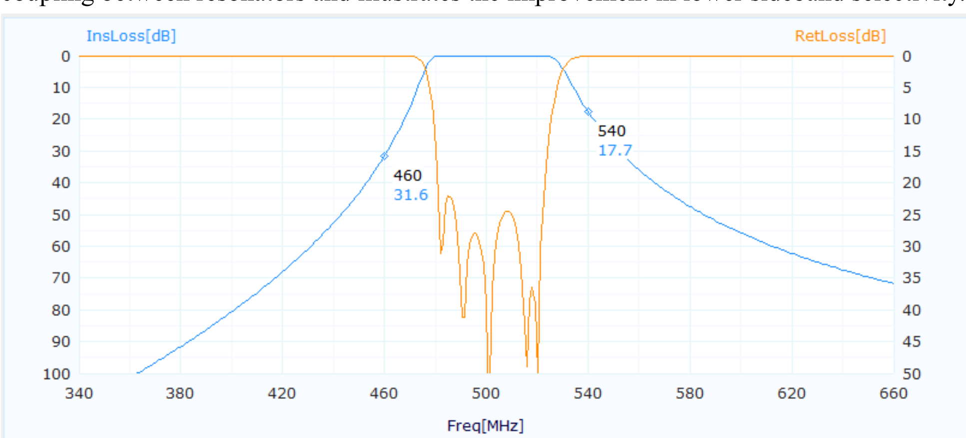
13.7.3.2. Solution 2 - Narrowband Microwave Filter solution from iFilter

A lesser-known passive filter design technique targets narrowband microwave filters, which is a significant portion of all filters used in high-frequency electronics. The design process starts by selecting an inverter-prototype, then applies frequency transformations to the shunt capacitors and replaces inverters by capacitive/inductive sections at the passband centre frequency, and finally applies impedance transformation. Although replacing inverters is an approximation, it invariably results in well-matched designs for narrowband filters.

The following design uses a Bandpass > Lumped > Narrowband Lumped Filter with an Inductively Coupled option. As the filter topology is a set of inductively coupled shunt resonator, the selectivity on the upper stopband is more pronounced than the lower stopband (see markers).



Various inductive/capacitive replacements are possible. The following graph illustrates a filter that employs only capacitive coupling between resonators and illustrates the improvement in lower sideband selectivity.



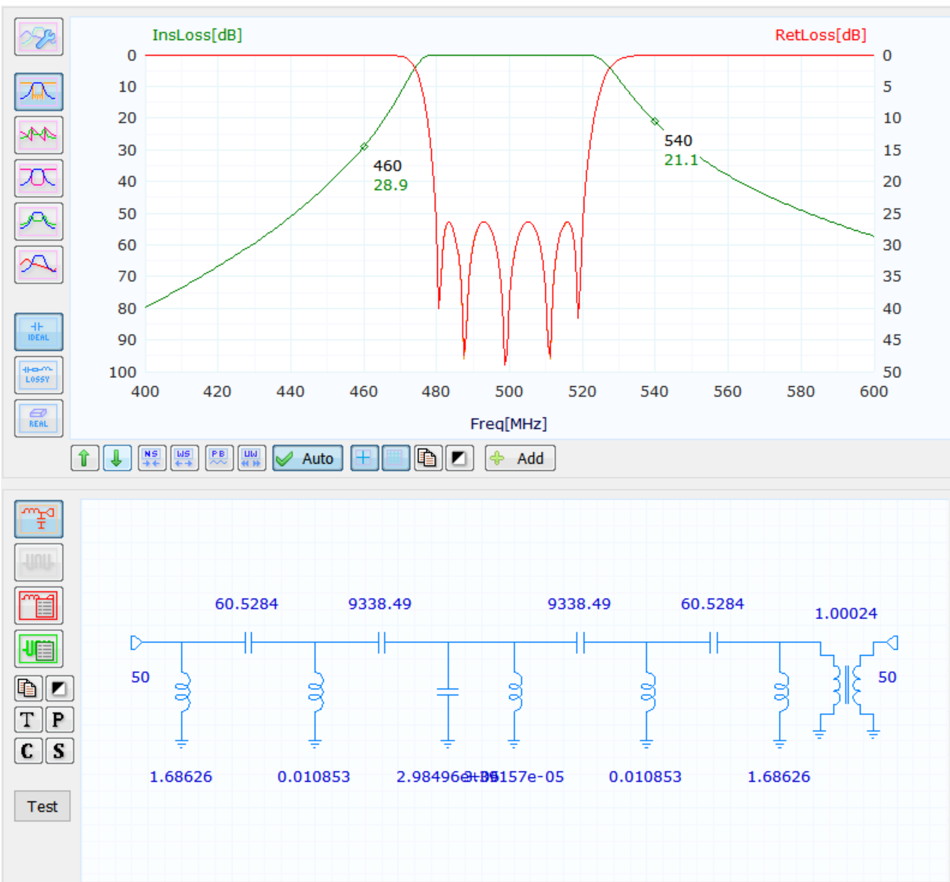
13.7.3.3. Solution 3 - Synthesis Solution from iFilter Synthesis

Using iFilter, solution 1 is an exact filter (the return loss behavior is exactly as prescribed in the original specification). Conversely, solution 2 (again using iFilter) is an approximate filter, yet close enough to the original specification, and one that possesses a topology that is realizable.

Solution 3, shown in the following design, uses iFilter Synthesis methods: TZ placement followed by Element Extraction. While the method yields an exact solution, it is not a design that is easily realized. iFilter Synthesis, however, also introduces equivalent circuit transformations that overcome this realizability issue.

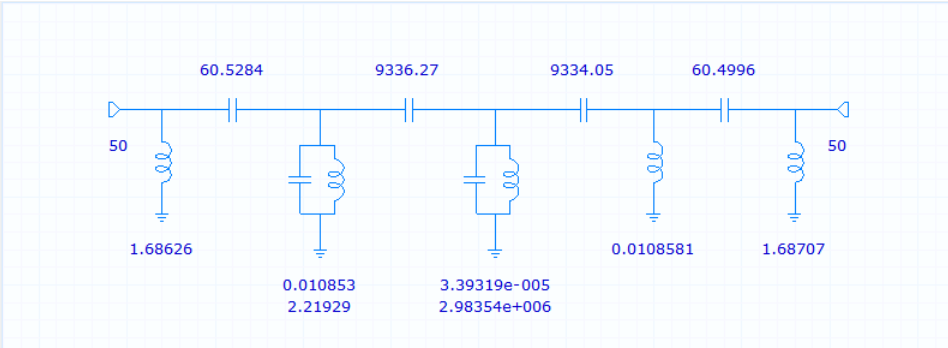
In solution 1, the bandpass filter has 5 TZs at f=∞ and 5 TZs at f=0. This results in perfect symmetry in the behavior of both sides of the passband for frequencies near the passband corners. As the frequency extends to zero and infinity, the symmetry is still maintained, although it is a geometric symmetry, which is difficult to visualize on a linear frequency plot of S21.

iFilter Synthesis allows the 10 TZs for this bandpass filter to be distributed unevenly. The following design uses a Bandpass > Lumped > Syn. Lumped Filter where 9 TZs are placed at f=0 and 1 TZ is placed at f=∞. Since there are more TZs at f=0, the filter is more selective in the lower stopband than in the upper stopband. This filter is exact, but the element values vary over a large range and there is a voltage transformer present at the load end with a 1:100024 turns ratio, so the filter is not particularly practical.

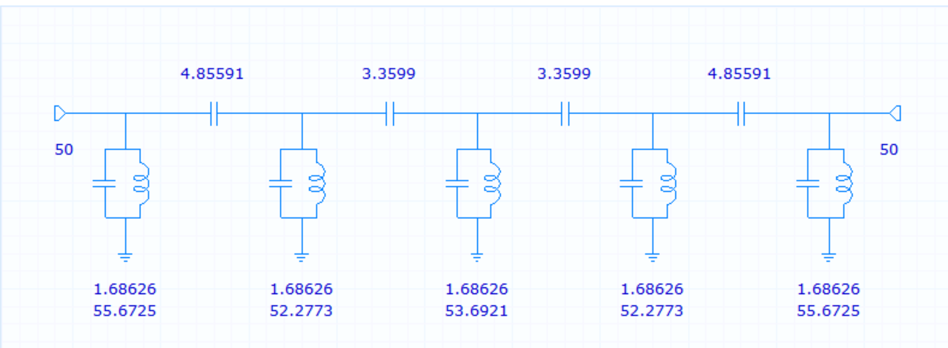


At this stage in the design flow, you can use Norton transformations to remove the unwanted transformer by canceling it with a transformer possessing an inverse turns ratio. To do so, create a new transformer one from within the elements by 1:1/1.00024 turns ratio using the Norton transformation. The series capacitor 9338.49nH and the shunt capacitor $2.98496*10^6$ nH form a capacitive L-Left section. You can replace this L-Left section with an L-Right section by using a specific transformation. iFilter Synthesis uses the terms L-Left, L-Right, Pi-, and Tee- to identify the circuit sections where successive Norton transformations can be applied.

The following circuit is obtained when the transformer is canceled after an L-Left to L-Right transformation is applied. It is important to note that the Norton transformations are exact, so the filter response does not change after applying them.



Although the transformer is removed by being absorbed by an inverse transformer, the filter is still not readily realizable given the large range of element values. The following variant, however, addresses this issue by applying an "All Equal Shunt Inductors" transformation. This results in a superior capacitively coupled bandpass topology, with a single inductance value (1.686nH) for the shunt inductors. With this realization, both the shunt and series capacitors possess a small range of values, which raises the possibility of simple tuning using printed elements. The final design has a practical topology where the passband return loss is exactly as initially prescribed.



As noted, there are many circuit solutions to a single filter specification. While standard iFilter provides several practical solutions, iFilter Synthesis provides more flexibility in the design process by being able to distribute the transmission zeros between DC and infinity and subsequently allowing the designer to apply various network transformations after element extraction to yield a more satisfactory solution.

13.7.4. Synthesis Process Flow

To use iFilter Synthesis effectively, you should know the filter synthesis process flow. The synthesis takes place in the following order:

- Place transmission zeros
- Extract circuit elements
- Apply circuit transformations

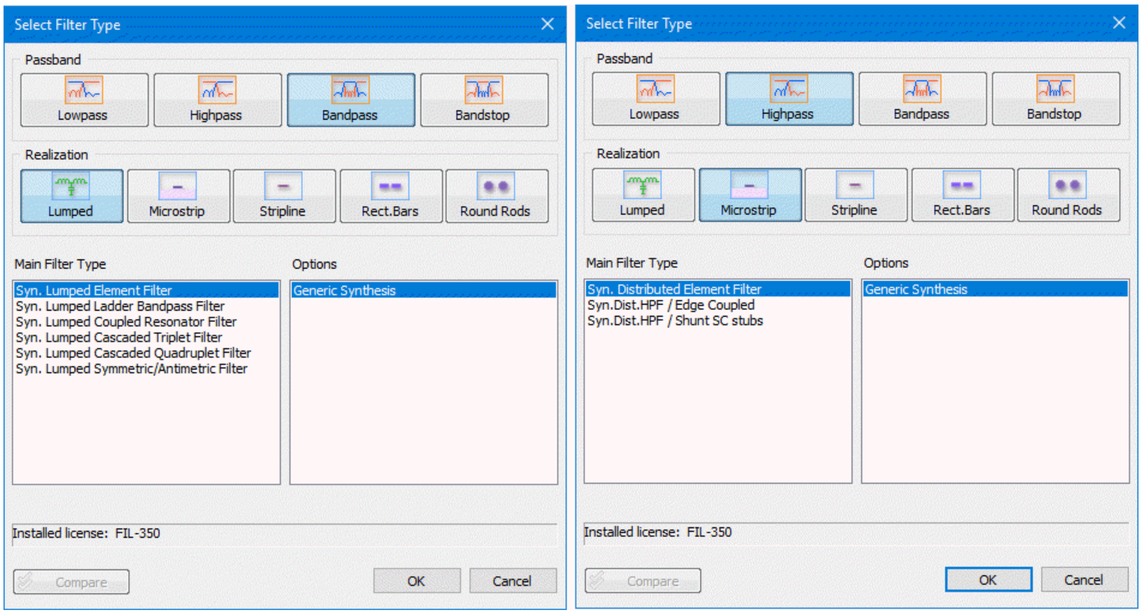
iFilter Synthesis provides manual or automatic control in any or all of these steps.

13.7.5. Designing in Manual or Semi-Automatic Mode

iFilter Synthesis has two main synthesis and design modes.

The first mode is manual or semi-automatic, and the second mode is fully automatic. You access these modes in the Select Filter Type dialog box at the beginning of the synthesis process.

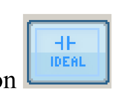
To run in manual or semi-automatic mode, select the option listed first under **Main Filter Type**. For lumped element filters, the filter type is named "Syn. Lumped Element Filter". For distributed element filters, the filter type is named "Syn. Distributed Element Filter". In both cases, there is a pre-selected single option listed under **Options** named "Generic Synthesis".



In manual or semi-automatic mode when you click **OK** the Advanced Synthesis dialog box displays.

13.7.6. Designing in Fully Manual Mode

To help understand the synthesis steps that are available, the bandpass filter described in the previous example is designed here using the fully manual mode. To constrain the example, the filter specification has a sideband attenuation of 30dB at 380MHz and 40dB at 595MHz.

Ensure that the units are set to MHz before defining the filter specification, then click the **Analyze Ideal** button 

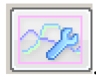
. The behavior of the filter when lossy and real elements are used is discussed in a later section.

The specification of the bandpass filter is

```
PB Ripple       0.01
Fo [MHz]        500
BW [MHz]        40
RSource/RLoad   50
```

Add two markers, the first at 380 MHz and the second at 595 MHz.

If the iFilter is not already in synthesis mode, click the **Select Design Mode** button  and select the **Synthesis** option to display the Select Filter Type dialog box. Alternatively, click the **Change Filter Type** button (at the top left; labeled with the current filter type) in the main iFilter dialog box.

In the Select Filter Type dialog box, select Bandpass > Lumped > Syn. Lumped Element Filter as a suitable manually synthesized filter for this example. Next you specify passband corners and passband ripple and then add markers to the insertion loss at this point by clicking the **Edit Chart Settings** button .

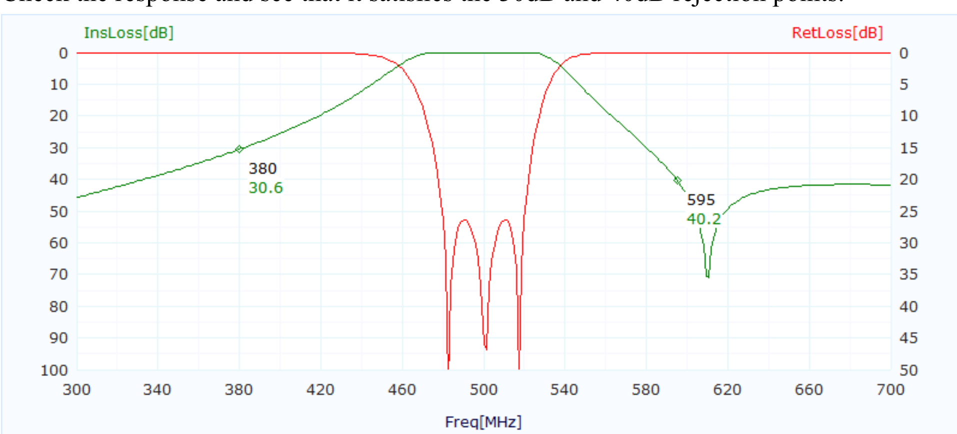
To place iFilter Synthesis in the fully-manual mode, in the Advanced Synthesis dialog box:

1. Clear the **Auto** check box under **Element Extraction**.
2. Select the **Apply Transformations** check box and **Custom** option under **Circuit Transformations**.
3. Do not use the **Prev** or **Next** solution buttons under **Automatic Extraction Solutions**.
4. Click the **Reset Extract** button under **Element Extraction** to clear any stored extractions.
5. Select **Type-B** under **Element Extraction** to start with series element.

The filter parameters are already specified in the main iFilter dialog box. The rest of the synthesis is completed by specifying and extracting transmission zeros:

1. Click the **Clear Transmission Zeros** button to start with a clean list.
2. Add 3 TZs at f=0 by clicking 3 times on the "**+**" button in the **ZERO** row.
3. Add 1 TZ at f=∞ by clicking once on the "**+**" button in the **INF** row.
4. Add 1 Finite TZ (FTZ) by clicking the "**+**" button next to the **Finite** listbox.
5. In the Add Finite TZ dialog box that displays, specify 610 MHz and then click **OK**.

Check the response and see that it satisfies the 30dB and 40dB rejection points.

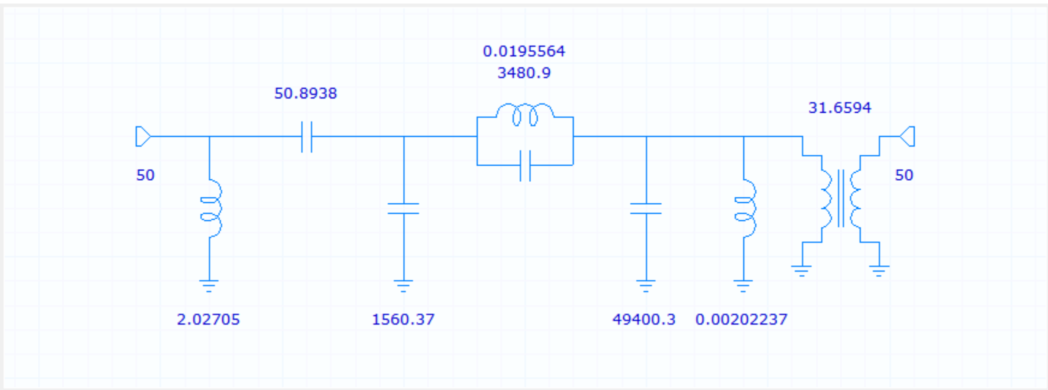


To extract the filter:

- Click the "**E**" button in the ZERO row to extract an element at f=0.
- Click the "**E**" button in the ZERO row to extract an element at f=0.

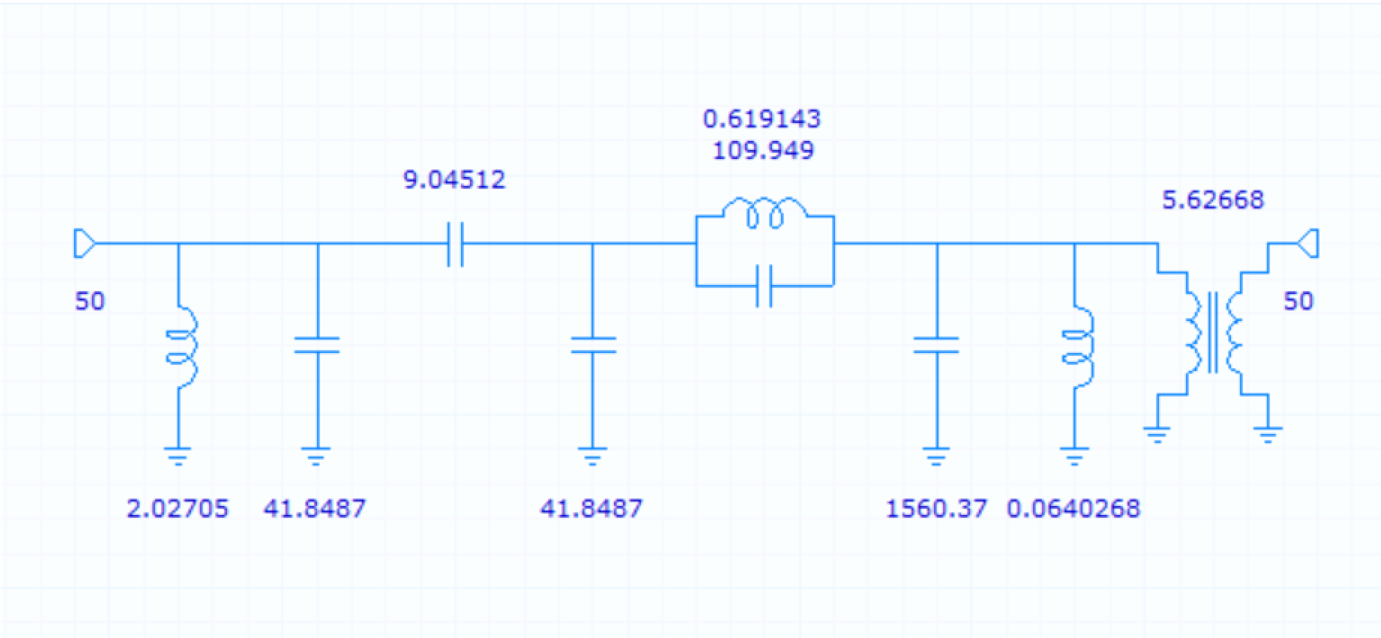
- Click the "**E**" button next to the **Finite** list to extract the element at f=610 MHz.
- Click the "**E**" button in the INF row to extract an element at f=∞.
- Click the "**E**" button in the ZERO row to extract an element at f=0.

You could extract elements in a different order. Although all of these filters have the response shown here, the resulting topologies and element values may differ considerably. The following circuit is obtained as a result of the defined extractions.

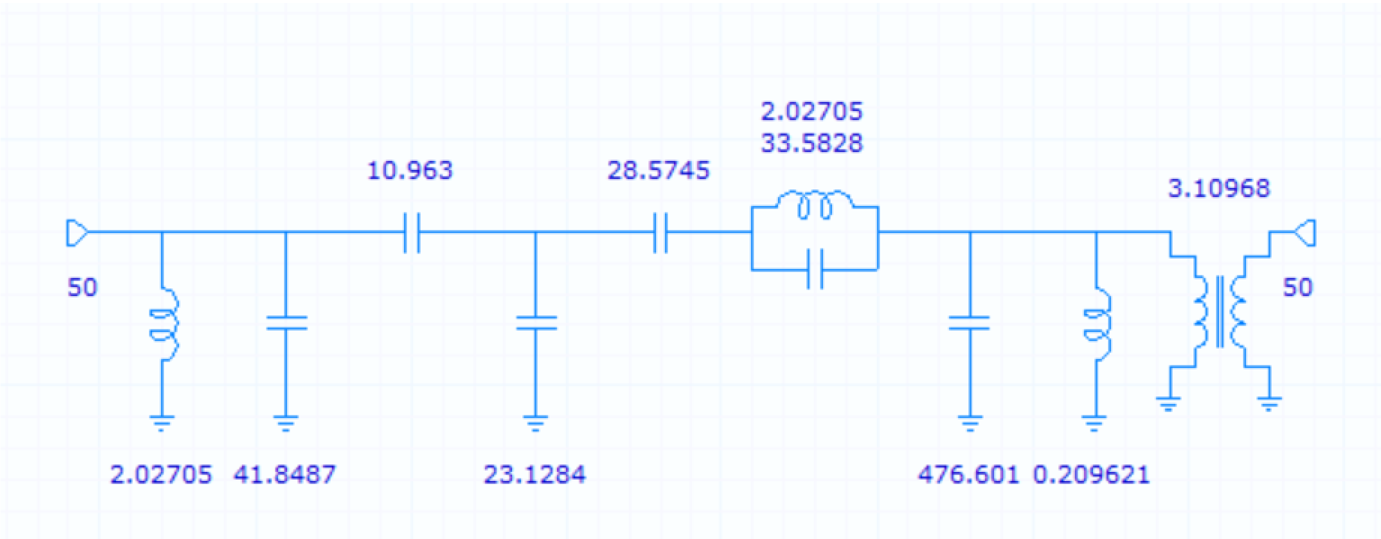


After defining the initial filter topology, you can now apply one or more Circuit Transformations. To do so, click the **Edit** button under **Circuit Transformations** to display the Circuit Transformations dialog box.

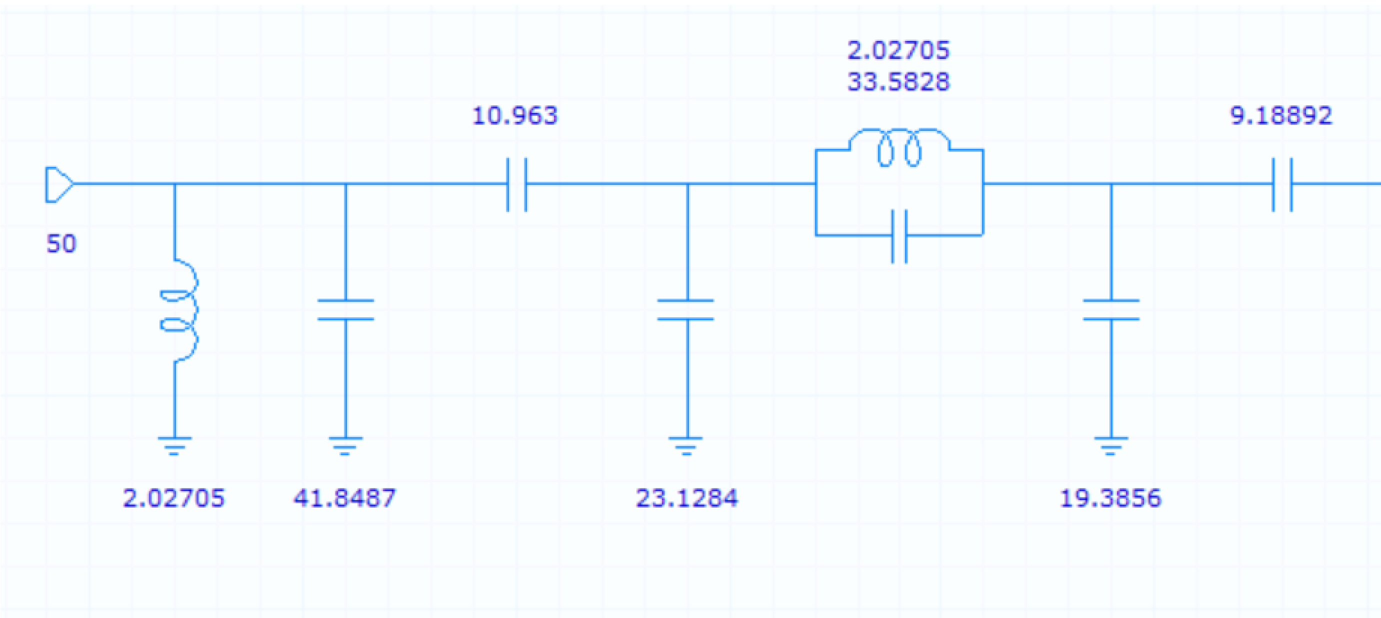
1. Select the first shunt capacitor (from the left), 1560.37pF.
2. Under **Available Transformations**, select the "LLeft to Pi - Symmetric Imp" transformation, then under **Apply Transformation To** select "CAP" and "Shunt" and enter "**1**" as the **Element Index**.
3. Under **Options**, select the **Simplify after transformation** check box.
4. Click the **Add** button, and then click the **Run List all** button to apply the selected transformation and change the circuit.



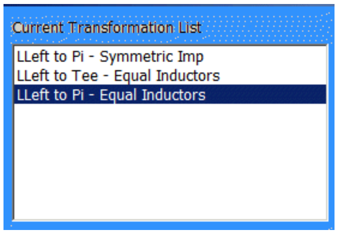
5. Select the second shunt capacitor, 41.8487pF.
6. Under **Available Transformations**, select the "LLeft to Tee - Equal Inductors" transformation, then under **Apply Transformation To** select "CAP" and "Shunt" and enter "**2**" as the **Element Index** (representing the 2nd shunt CAP).
7. Under **Options**, select the **Simplify after transformation** check box.
8. Click the **Add** button, and then click the **Run List all** button to apply the selected transformation and change the circuit.



9. Select the third shunt capacitor, 476.601pF.
10. Under **Available Transformations**, select the "LLeft to Pi - Equal Inductors" transformation, then under **Apply Transformation To** select "CAP" and "Shunt" and enter "**3**" as the **Element Index** (representing the 3rd shunt CAP).
11. Under **Options**, select the **Simplify after transformation** check box.
12. Click the **Add** button, and then click the **Run List all** button to apply the selected transformation and change the circuit.



The following figure shows the list of applied transformations.



You can see that this filter design is a realizable filter. Most notably all the inductors have a single value, that is 2.027nH. A 2nH quality RF inductor in 0402 or 0603 sizes can be selected or the inductors can be wound by hand. Capacitors range from 9.2 to 43.1pF and they are readily found in MLCC capacitor toolkits.

13.7.6.1. Tuning the Finite TZ

Any time during the synthesis, even after all the transformations are applied, you can tune the Finite TZ at 610MHz to move the rejection point along the frequency axis, perhaps to improve a passband slope or accommodate a late change in the filter specification. To do so, first select the FTZ from the list, then click in the **FTZ** text box and move the mouse wheel up and down to change the value. All the design steps that are integrated into the filter design up to this point in the design flow are repeated with the new FTZ value.

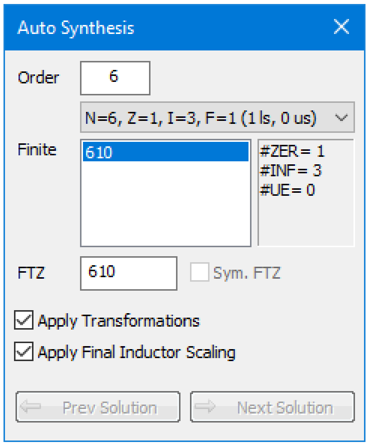
13.7.7. Designing in Semi-Automatic Mode

To design the bandpass filter in the previous example in the semi-automatic mode, you specify the passband center and bandwidth, and the TZs in the same way as the manual mode. Next you click the **Next** and **Prev** buttons under **Automatic Extraction Solutions** until you see solution 1 of 5. This solution is programmed as a built-in Ladder solution for this particular set of TZs. The result is the same circuit that was obtained by the tedious method in the fully-manual mode. Bandpass filters are commonly used, so 450 solutions are programmed into the iFilter Synthesis. Overall, there are about 1500 solutions in the wizard.

13.7.8. Designing in Fully Automatic Mode

About 1500 variations of TZ placements, extraction sequences, and transformations are programmed into iFilter Synthesis for designers. You can access them from the Advanced Synthesis dialog box as well as the Auto Synthesis dialog box.

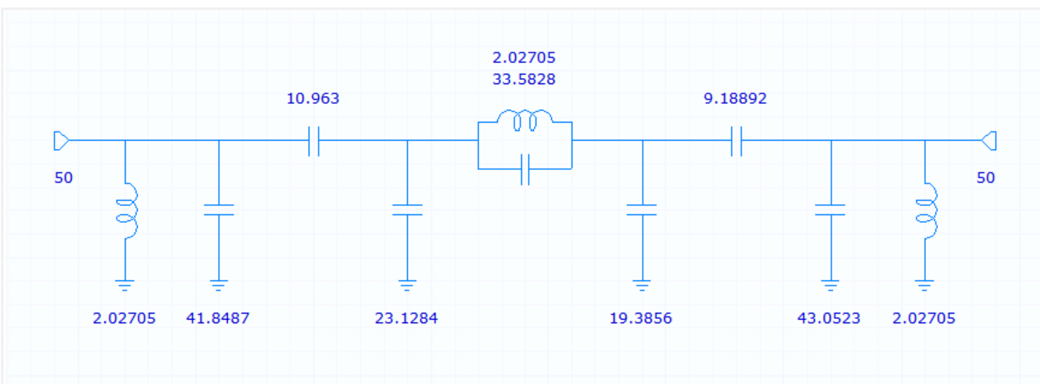
To access the Auto Synthesis dialog box, in the Select Filter Type dialog box under **Main Filter Type** select any filter type other than the first option and then click **OK**. For example, selecting the filter described in the previous examples (Bandpass > Lumped > Syn. Lumped Ladder Bandpass Filter) displays this dialog box.



1. To design the same example filter, specify "6" as the **Order**.
2. Select "N=6, Z=3, I=1, F=1 (0 ls, 1 us)" from the drop-down box.
3. Select and change the somewhat-arbitrary FTZ to 610.
4. Select the **Apply Transformations** check box.

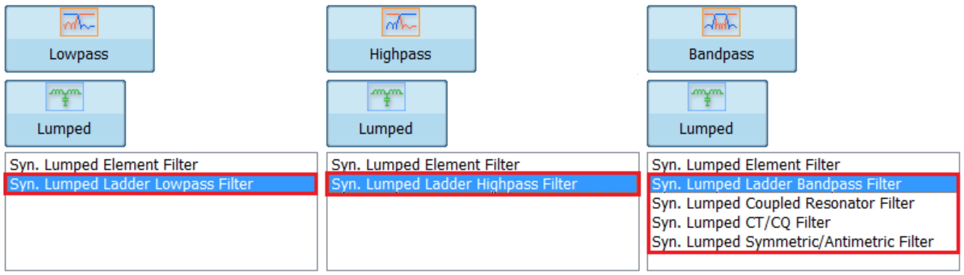
Since there is only one solution for the Syn. Lumped Ladder Bandpass Filter type, the **Prev** and **Next** buttons are disabled.

The following figure shows the resulting filter.

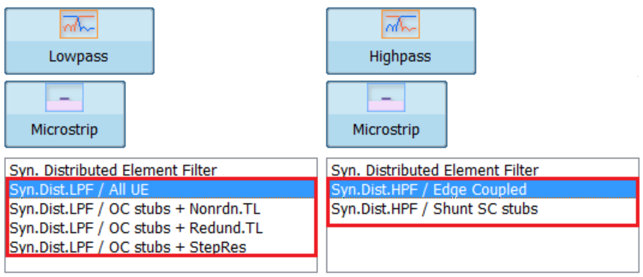


Note that this is the same filter obtained in the Manual mode because the transformations stored in the Syn. Lumped Ladder Bandpass Filter is in the same order that was manually specified.

Most common topologies are accessible through this semi-automatic mode. Other than the first filter type in the list, all other filter types listed in the Select Filter Type dialog box have predefined topologies. The lumped versions are shown in the following figure.



The distributed filters with pre-programmed topologies are shown in the following figure.

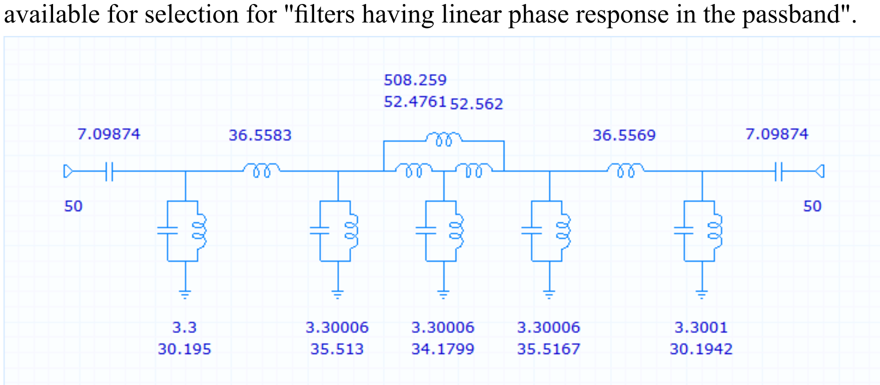


As new topologies become available, they will be added to the list. Note that the first filter type in these lists are manually extracted filters for the Advanced Mode.

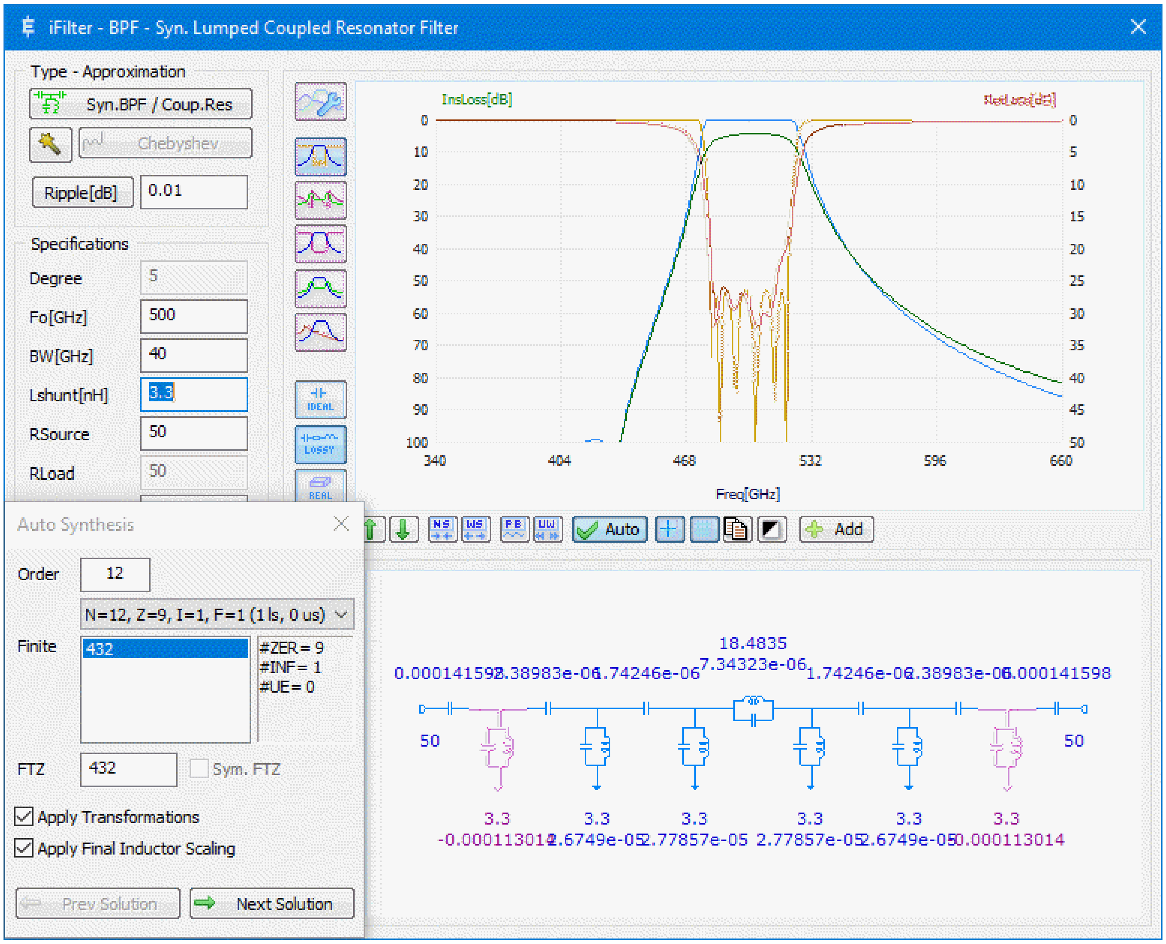
13.7.9. iFilter Synthesis Features

The following list highlights some of the important capabilities available within iFilter Synthesis. This list is not comprehensive.

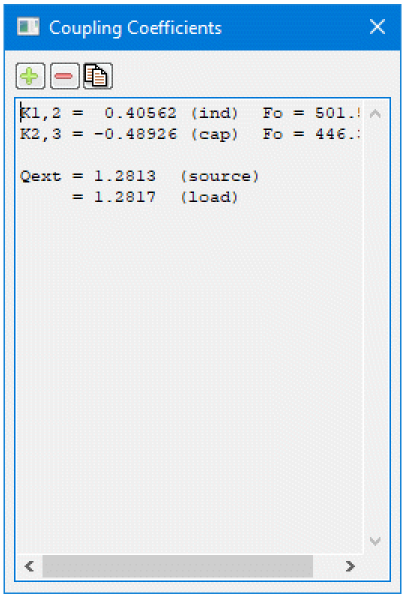
- Lumped Bandpass filters contain a CT/CQ option. These are cascaded triplets and quadruplets which provide cross-coupling within a ladder structure. While there are exact CT/CQ sections, there are also approximate sections available for selection for "filters having linear phase response in the passband".



- Lumped Bandpass Coupled Resonator filters feature all-equal shunt inductors which can be specified directly from the main iFilter dialog box. For example, the following filter only has 3.3nH shunt inductors. Having identical inductors means a reduced Bill of Materials.



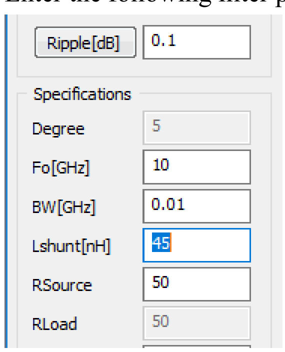
- In distributed filters, you can add and extract contributing unit elements in iFilter Synthesis. In a non-contributing UE filter, 50-ohm transmission lines are inserted from source and load ends and moved in towards the mid-circuit by applying Kuroda transformations. In the contributing unit element case, the same topology can be obtained by synthesizing unit elements directly within structure. This feature is used in “Open Circuit Stubs with Non-redundant Transmission Lines” design, which you select by choosing the Lowpass > Microstrip > Syn. Dist. LPF - OC stubs + Nonrdn. TL filter in the Select Filter Type dialog box.



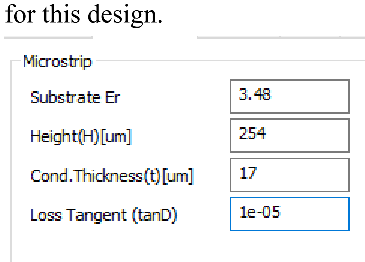
13.7.10. Distributed Element Lowpass Filter Example

This example shows the design of 7th order 10 GHz microstrip lowpass filter in iFilter Synthesis.

1. In the Select Filter Type dialog box, select the Lowpass > Microstrip > Syn. Distributed Element Filter, then click **OK**.
2. Enter the following filter parameters in the main iFilter dialog box.



3. Finish the design setup by clicking the **Design Options** button and on the **Technology** tab of the Distributed Model Options dialog box that displays, enter the substrate parameters. Use "0.010" (0.254mm) Rogers RO4350B substrate for this design.



4. In the Advanced Synthesis dialog box under **Element Extraction**, clear the **Auto** check box to enable manual extraction, then click the **Clear Transmission Zeros** button above it to do a clean start.

13.7.10.1. Lowpass Filter with Monotonic Stopband

Filters with no finite transmission zeros have a monotonically increasing attenuation in their stopband. At the far away frequency from the passband, f=INF, no transmission occurs, i.e. S21=0.

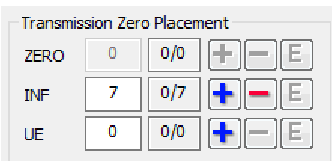
For lumped lowpass filters, INF occurs at f=infinity Hz. For distributed filters, INF occurs at multiples of quarter wavelength frequency, Fq. For lowpass filters, Fq is related to Fp in the following equation: Fq = Fp * 90/EL

So for EL=45 deg, and Fp=10 GHz, Fq is found as 20 GHz.

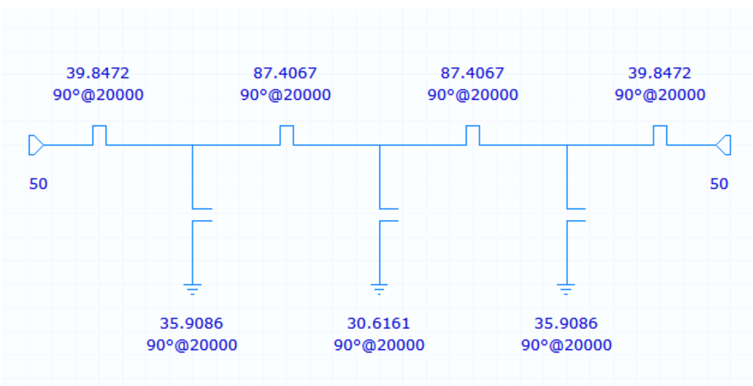
13.7.10.2. Solution #1 – All Transmission Zeros located at Fq

As the first variation, you design with 7 TZ all located at infinite frequency. Each TZ at infinite frequency adds 1 order to the filter.

To place 7 TZ, click the "**+**" button in the INF row.



Next extract the element values. Since you only have all the TZs at f=INF, the only way to extract TZs is to click the "E" button in the INF row 7 times.



Note for future reference that the topology is symmetric around the middle shunt stub (30.62ohms).

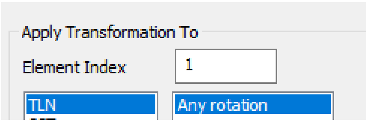
Initially, this looks like a lumped element filter, where the series inductors are replaced with short-circuited stubs, and the shunt capacitors are replaced with open-circuited stubs. Instead of having inductances and capacitances, the stubs have impedance and lengths, which are quite reasonable. However, there is a fundamental problem with the structure: Series short-circuited stubs cannot be realized on microstrip, so an equivalent circuit that is realizable must be found.

Kuroda transformations are the most common way of converting series short-circuited stubs into shunt open-circuited stubs. To do so, you apply a series of Kuroda transformations until all stubs are replaced:

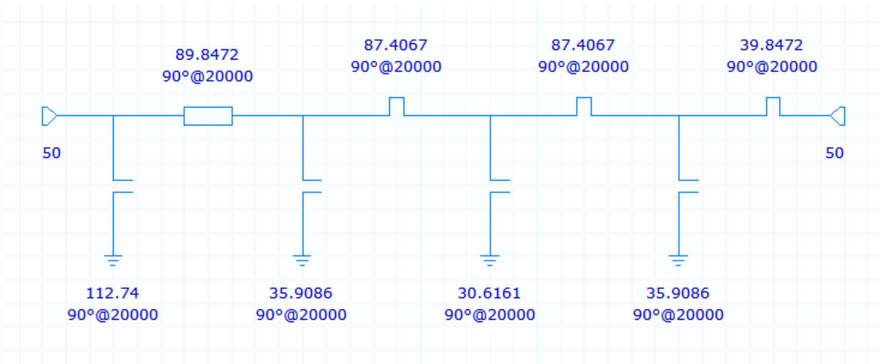
1. Under **Circuit Transformations**, select the **Apply Transformations** check box, click the **Custom** button, and then click the **Edit** button to display the Circuit Transformations dialog box.
2. In the **Available Transformations** list, select "Add Transmission Line to Source Side" and click the **Add** button to add it to the transformation list. Adding 50-ohm transmission lines to source and load sides does not change the response.

3. Click the **Run List all** button.
4. In the modified circuit, select the transmission line on the left.
5. In the **Available Transformations** list, select "Kuroda Right – Full Stub" and click the **Add** button to add it to the transformation list.
6. Click the **Run List all** button.

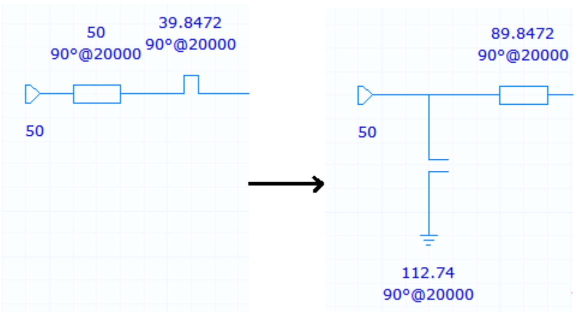
Note that the options related to the selected commands must be set correctly while adding to the list. Each command has a different options setting. For the software to apply the Kuroda transformation, it should know which transmission line is intended. You should specify the FIRST transmission line for this transformation by entering "1" in the Element Index box as shown in the following figure.



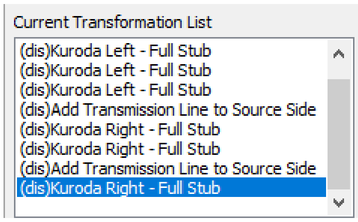
The following circuit results:

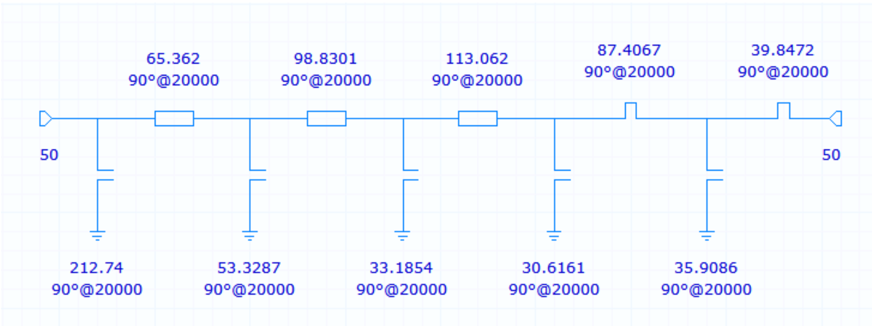


Kuroda transformation turned a "Transmission Line + Series SC Stub" into "Shunt OC Stub + Transmission Line" section. These two filter sections have identical frequency response. Similarly, all Kuroda transformations yield identical equivalent circuits; they are exact transformations.



By continuing to insert transmission lines and applying Kuroda transformations, all series SC stubs can be transformed into shunt OC stubs, although it is a tedious process. You should move the first transmission line towards the right by 3 successive "Kuroda-Right - Full Stub" transformations. You then add an "Add Transmission Line to Source Side", and apply 2 successive "Kuroda-Right - Full Stub" transformations as well. Finally, you add one more "Add Transmission Line to Source Side" and a "Kuroda-Right - Full Stub" transformation.



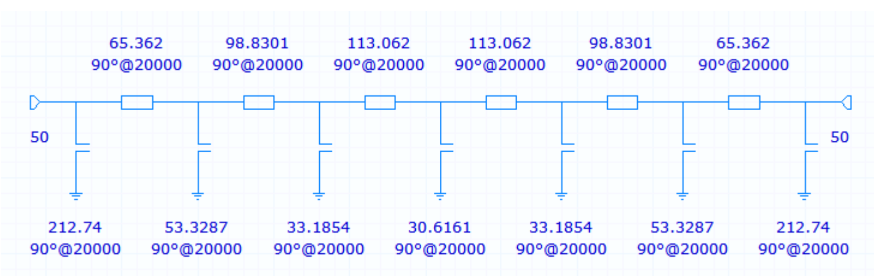


You can now continue to add transmission lines and do Kuroda transformations, however, it is tedious to do it from the left-hand (Source) side. You can also add transmission lines to the Load side and apply "Kuroda-Left - Full Stub" to them, however you must add 9 more transformations.

Short Cut 1

Alternatively, there is an easy way of converting the stubs on the right-hand side. As previously noted, 30.62-ohm shunt stub is the original center (pivot) of the original symmetric filter. You can now mirror the circuit around that stub, which is the same as repeating all the transformations on the right-hand side. Note that you should set the **Element Index** to "4" before adding the command to the list. If the command is selected first and stub is selected after, the wizard already places the correct index into the box.

After adding "Mirror Circuit" to the **Current Transformation List** and running the whole list, the following all-shunt-OC filter results:



Short Cut 2

The schematic in Short Cut 1 is an almost-realizable topology on microstrip. The only exception is the first shunt stub, whose impedance is 212.74 ohms. It’s difficult to realize impedances above 150 ohms, as the line widths become too small. The rest of the topology is suitable for construction, so at this stage for realization you can only replace the high impedance lines with lumped inductors, or try another solution.

13.7.10.3. Solution #2 – Filter with Non-redundant Transmission Lines

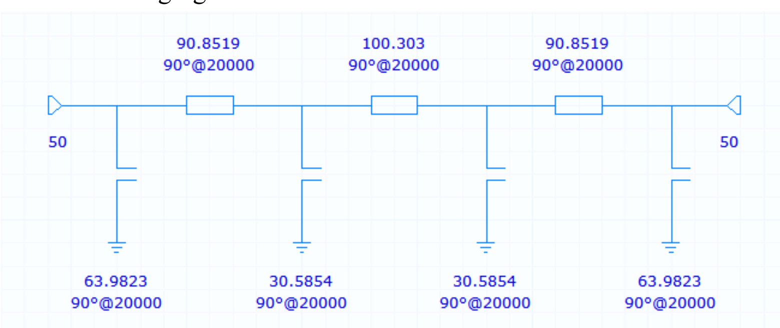
In Solution 1, you ended with series transmission lines which were not there in the original extracted circuit. These transmission lines are inserted and shifted through Kuroda transformations, so they are redundant elements (they don't directly contribute to the filter selectivity) and do not count towards the filter order.

In this solution, you use transmission lines that contribute to the filter selectivity by adding a set of TZs in the following form:

- 4 INF
- 3 UE

Each UE practically adds 1 TZ effect to the filter response, so effectively you now have a 7th order filter. To extract the element values click the "**E**" buttons in the following order: INF-UEL-INF-UEL-INF-UEL-INF

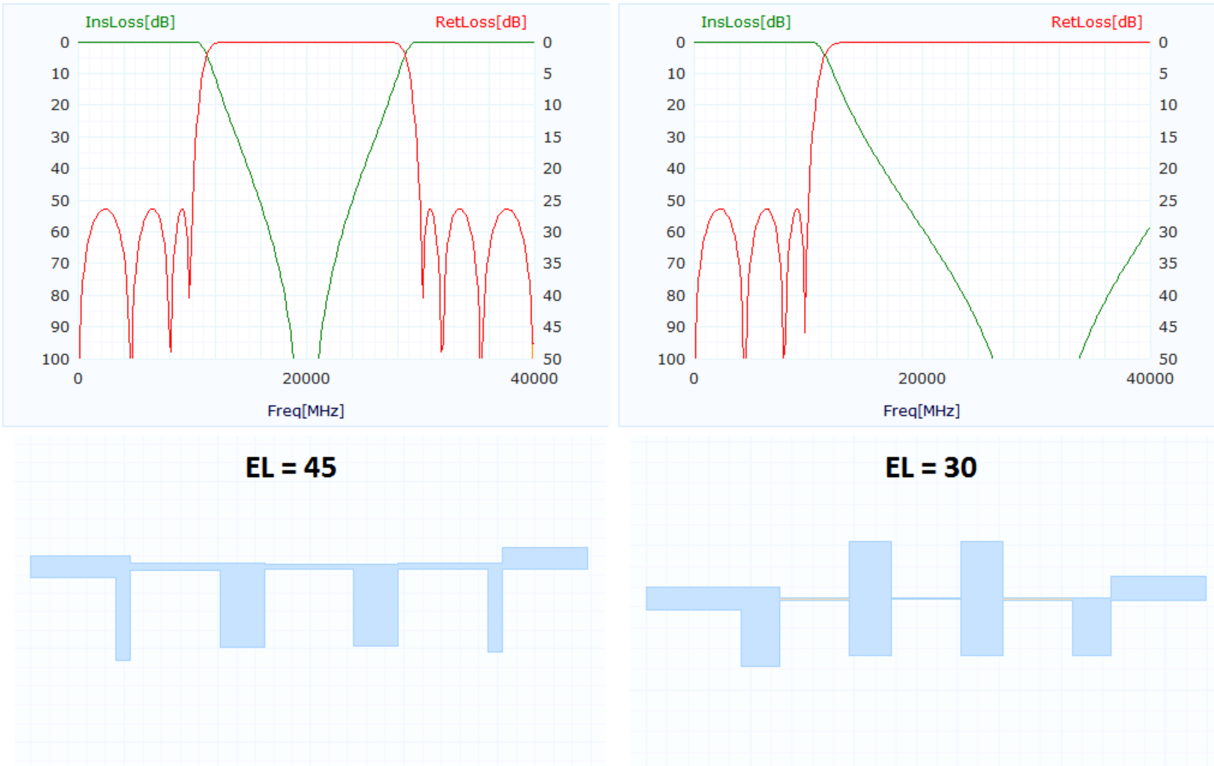
Without performing any circuit transformations, you have a realizable topology and reasonable element values, as shown in the following figure.



The layout for this filter is shown in the following figure.

As an alternative extraction, under **Element Extraction** you can select the **Auto** check box. iFilter Synthesis matches the transmission zero list with pre-stored templates and uses the corresponding extraction sequence for this common template, (n+1) INF + n UE.

At this stage, you can investigate how the stopband can still be manipulated without touching the transmission zeros. As noted previously, the electrical length parameter specifies the length at the passband corner. When it is 45-degrees, f=INF=Fq occurs at 2*Fp. You can set it to a smaller value, like 30-degrees, and Fq is pushed higher in frequency to 30 GHz. The effect of changing EL from 45- to 30-degrees is shown in the following figure.



13.7.10.4. Solutions with Finite TZs

When finite TZs (FTZ) are considered, stopband attenuation can be formed to provide infinite attenuation at selected frequencies. Every FTZ adds 2 orders to the filter, so 2 INF can be replaced with a single FTZ. This swap changes the slope of attenuation near f=INF and (pull) S21 around the finite TZ to zero. Placing a FTZ is similar to pressing the middle of a balloon that is fixed between two points: when pressed, it bubbles up more on two sides further up.

For a 7th order filter, the following sets of transmission zeros are possible:

- 7 INF
- 6 INF, 1 UE
- 5 INF, 2 UE
- ...
- 1 INF, 6 UE
- 7 UE
- 5 INF, 1 FTZ
- 4 INF, 1 UE, 1 FTZ
- 3 INF, 2 UE, 1 FTZ
- ...
- 3 INF, 2 FTZ
- 2 INF, 1 UE, 2 FTZ

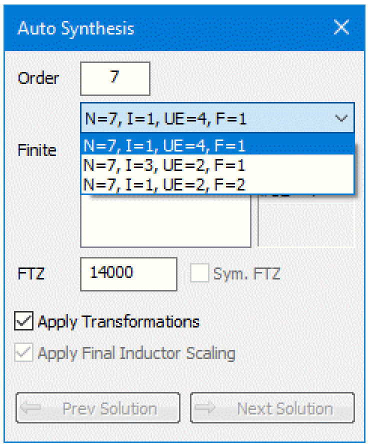
- ...
- 1 INF, 3 FTZ

Each of these TZ sets can be investigated to see if they are realizable, although with the number of combinations it would be very time consuming.

In iFilter Synthesis, some of the most realizable topologies are pre-stored as solution templates. Each solution has its own TZ extraction sequence and list of transformations that are applied after extraction. The following sections include solutions found among those pre-stored templates.

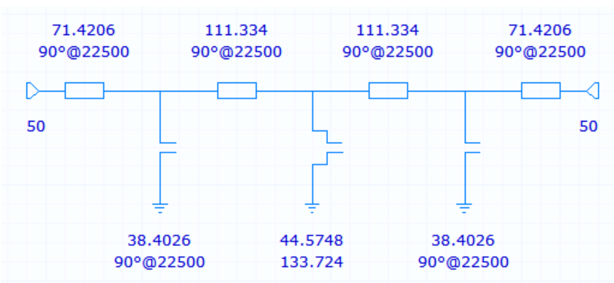
Solution #3 – Filter with 1 TZ at Inf, 4 UE and 1 FTZ

To access this solution, in the Select Filter Type dialog box select the Lowpass > Microstrip > Syn.Dist.LPF/OC stubs + Step.Res. filter type and click **OK** to display the Auto Synthesis dialog box.



Specify "7" as the **Order**, then click in the drop-down box to view three 7th order filter options. Select the first option with 1 TZ at INF, 4 UE and 1 FTZ.

You can tune the location of finite TZ by selecting it from the **Finite** list and entering a new **FTZ** value (alternatively, scroll the mouse-wheel up and down). The following figure shows **FTZ** specified as 15 GHz.



Note that the middle element is a step resonator. It consists of two cascaded transmission lines connected to the main filter arm as a shunt element. The final end of 133.72 ohms is left open.

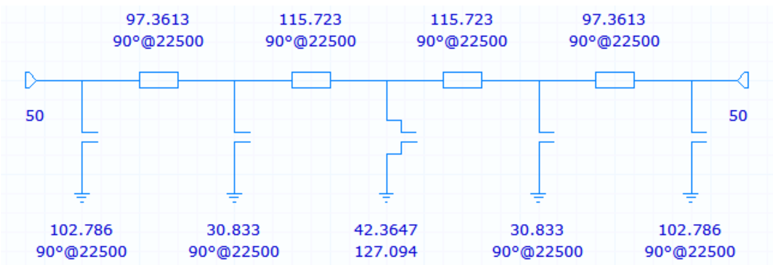
To see the extraction sequence and transformations applied to obtain this circuit, click the "**S**" button in the main iFilter dialog box to display the Synthesis Information window. The first few lines from the window summarize the actions performed:

```
Extraction Order:

Type A: U U F1 I U U
Solution: 1 / 1

Transformations:
# 1: Kuroda to All Series SST (1 at 1)   [OK]
# 2: All Series Stub Res to 2-step Resonator (1 at 1)   [OK]
```

If the steps are followed manually in the Advanced Synthesis dialog box, the same filter is obtained, however the Auto Synthesis mode makes it much easier by doing everything automatically.

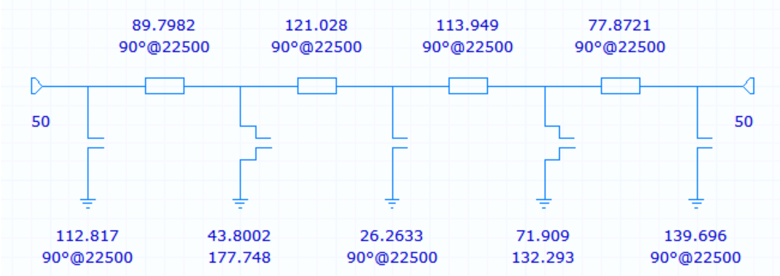
Solution #4 – Filter with 3 TZ at Inf, 2 UE and 1 FTZ

Select the second option of the three 7th order filter options in the drop-down box, with 3 TZ at INF, 2 UE and 1 FTZ. The procedure is almost the same as solution #3, however the end topology is slightly different as there are two extra shunt stubs at the termination ends.



Solution #5 – Filter with 1 TZ at Inf, 2 UE and 2 FTZ

The third option of the three 7th order filter options in the drop-down box gives the following topology (FTZ1 tuned to 13.4 GHz, FTZ2 tuned to 15.9 GHz for equiripple-like stopband).



13.8. Impedance Matching Wizard (iMatch)

The Cadence® iMatch™ impedance matching wizard (iMatch) uses the Cadence iFilter™ filter synthesis wizard interface. Starting, running, and closing the iMatch Wizard are similar to the same operations with the iFilter Wizard. For detailed information, see “iFilter Filter Wizard”.

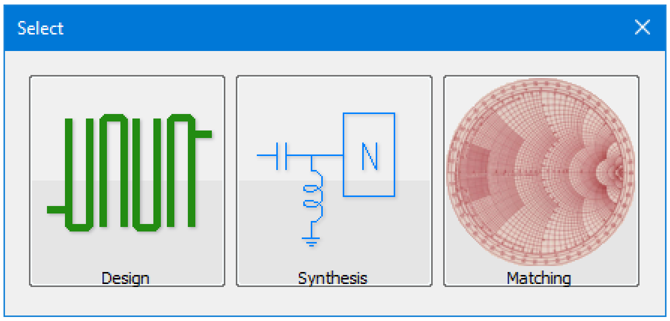
This wizard displays in the Cadence AWR Design Environment® platform if you have the proper license file (FIL-350, FIL-300, or FIL-050) to run the wizard.

13.8.1. Using the iMatch Wizard

iMatch can run as a stand-alone license or as an integrated feature with an iFilter license. In a stand-alone configuration, the iFilter Wizard can only design impedance matching type circuits. In an integrated configuration, the iFilter Wizard recognizes iMatch circuits as a special filter type.

13.8.1.1. Running the iMatch Wizard

You can run the iMatch Wizard to create a new impedance matching network or to modify an existing impedance matching network. To create a new matching network, open the **Wizards** node in the Project Browser and double-click the **iFilter Filter Wizard** then click **Matching**.



You can edit terminations, specifications and matching options. After every change, the wizard recalculates the values, redoes the realization (layout or part selection) and calculates and plots the response. You do not need to press a special button after modifications as all the views are kept current.

To enter a different value or specification in the Matching dialog box you can use the keyboard, click the up/down arrows next to an option to increase/decrease values, or use the mouse wheel (click in the desired edit box and scroll the mouse to increase/decrease the value). The step size is automatic based on the type and value of the edit box. Press **Ctrl** while scrolling to increment/decrement with a smaller step size.

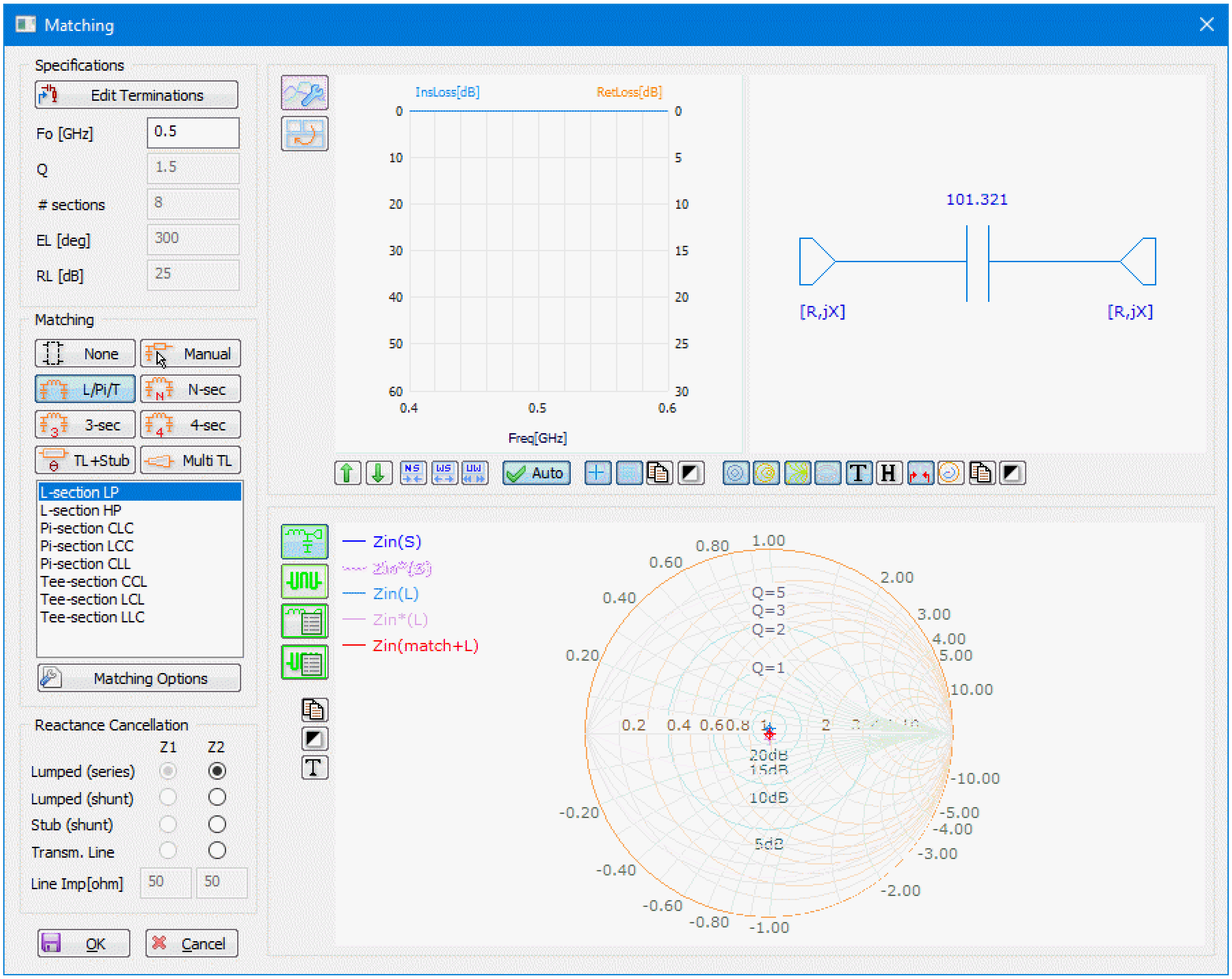
13.8.1.2. Closing the Wizard

To close the iMatch wizard:

- Click **OK** in the Matching dialog box to save your design.
- In the main iFilter dialog box, click **Generate Design** to create a schematic, graph(s) and other items in the AWR Design Environment platform.
- Click **OK** to create a filter design item under the **iFilter Filter Wizard** node in the Project Browser only. No schematic, graph(s) or other items are created. This is the only way to save a wizard state for later reuse.
- Click **Cancel** to close the Matching dialog box without saving.

13.8.2. iMatch Wizard Basics

The Matching dialog box (main iMatch dialog box) as shown in the following figure, is comprised of specifications on the left-hand side and graphics on the right-hand side. Graphics include the insertion loss and return loss graph, Smith Chart, and schematic/layout drawing.



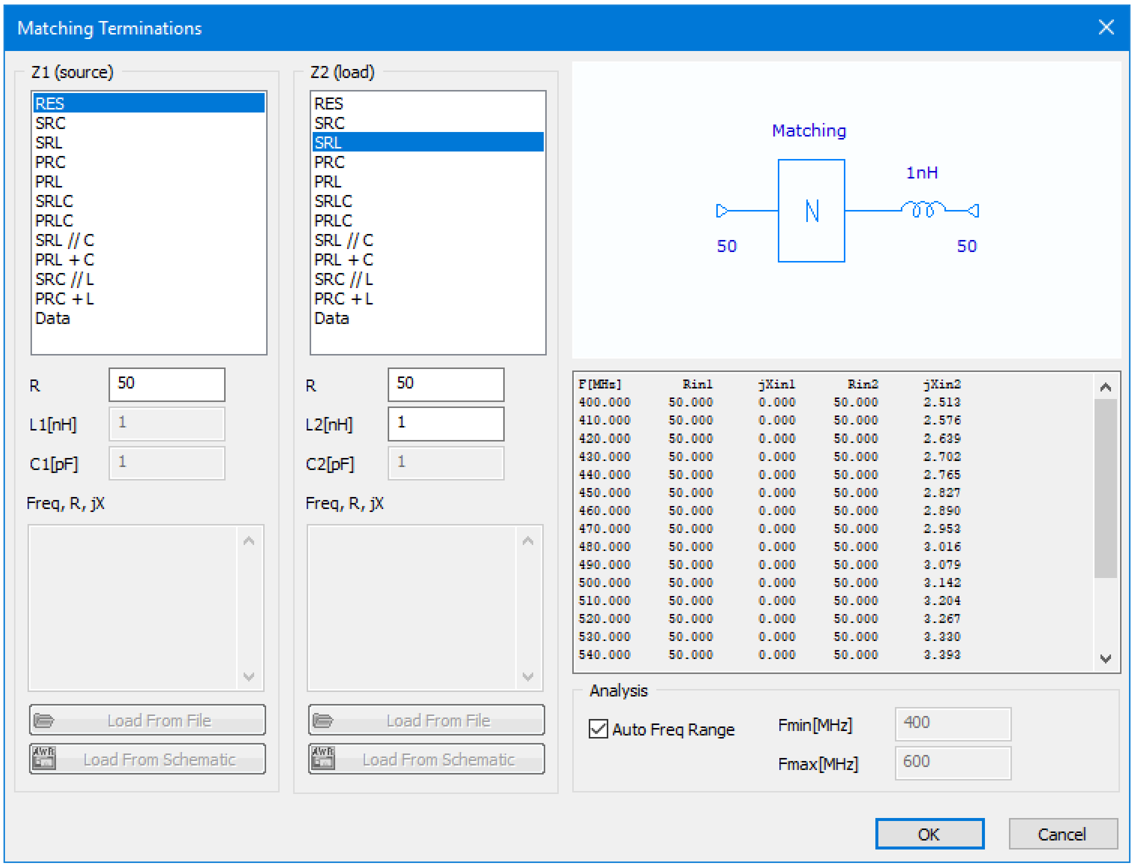
To create a typical iMatch design:

1. Click the **Edit Terminations** button to specify load and source terminations in the Matching Terminations dialog box, then click **OK**.
2. To select additional options, click the **Matching Options** button to display the Matching Options dialog box.
3. In **Fo**, enter the center frequency of the matching network.
4. Click one of the six buttons in the **Matching** group to select the matching type. See "Impedance Matching Types" for information on the matching types.
5. Click the **Matching Options** button and select an option in the Matching Options dialog box.
6. Select a **Reactance Cancellation** method for terminations. See "Reactance Cancellation" for more information on the available options.
7. In **Q**, **# seconds**, and **EL [deg]**, enter further design specifications if available.
8. Repeat steps 3 - 7 to obtain the most appropriate design.

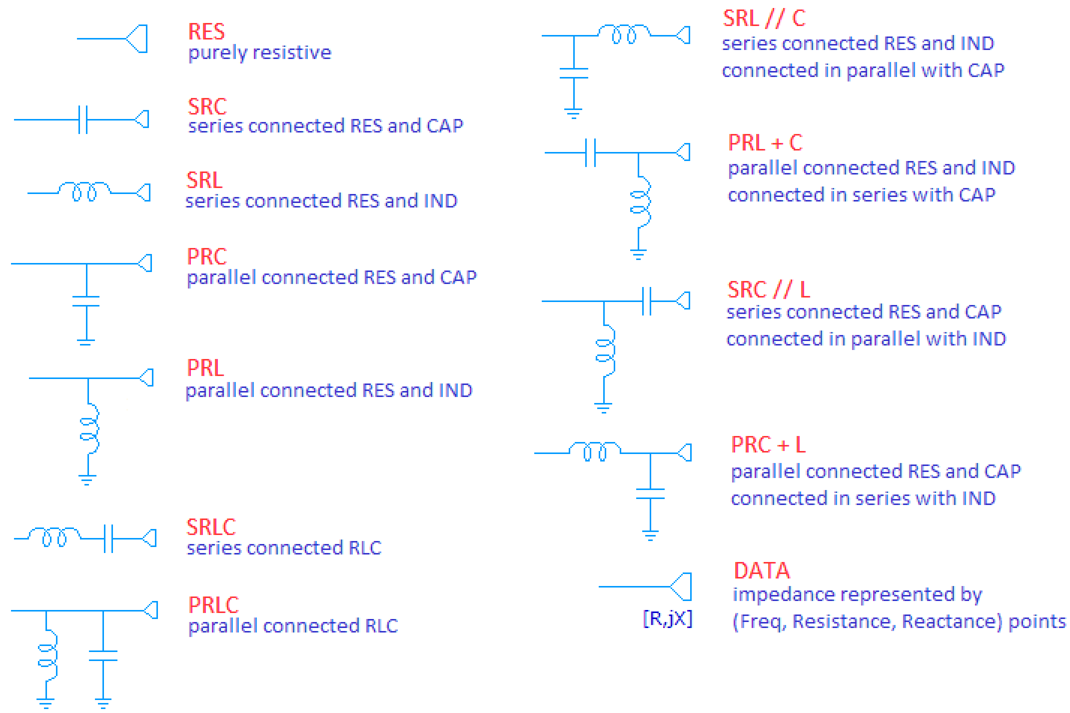
9. Click **OK** to save the current design, close the dialog box and transfer the data to the main iFilter dialog box. You can make further adjustments such as specifying the **Fo**, or selecting technology parameters in this dialog box.
10. Reopen the Matching dialog box as required from within the main iFilter dialog box.
11. Click the **Generate Design** button in the main iFilter dialog box to generate the design (schematics, graphs, and data) in the Cadence Microwave Office® software Project Browser.

13.8.2.1. Matching Terminations Dialog Box

The Matching Terminations dialog box is used to specify source and load impedances (terminations) of the impedance matching design. To access this dialog box, click the **Edit Terminations** button in the Matching dialog box.



Specifications are grouped on the left side of the dialog box. Specifying terminations is identical for source and load. A termination can be represented by a single element, a combination of elements, or a frequency-impedance data array. Major passive elements (RES, IND, and CAP) and their various combinations are available for selection. After selecting a model for the termination, enter the R, L or C values if they are enabled based on your selections. On the right side of the dialog box, a representative schematic and selected frequency information are shown for reference. The following termination types are available in iMatch:



Some devices that require matching are represented by impedance vs. frequency. In this case, select **Data** from the **(source)** or **(load)** box and enter the data in the **Freq, R, jX** box. You can specify Data as frequency, real, and imaginary parts of impedance, with comma delimiters, for example:

100, 45, 5

150, 50, 7

200, 55, 9

This means 45+j5 Ω at 100MHz, 50+j7 Ω at 150MHz, and 55+j9 Ω at 200MHz. You can use the following means to enter data in the **Freq, R, jX** box:

- Manually enter the data
- Click the **Load From File** button to load data from a *.S1P* or *.S2P* Touchstone format file. If the file is *.S1P* format, then the impedance is given in the file. If the file is *.S2P* format, you can use either S11 or S22 as the matching impedance. This choice is presented in a simple dialog box: Use S11 for input impedance? Click NO to use S22. Click YES to use S11, NO to use S22 for the impedance.
- Click the **Load From Schematic** button to load data dynamically from Microwave Office schematics. A Select Schematic and Port dialog box displays to allow you to select an Microwave Office schematic in the current project. The schematics list contains schematics with only 1 or 2 ports. Select **Port 1** to use this port to calculate input impedance towards the matching network (the iMatch wizard calculates the input impedance starting from the first element after port 1 looking into port 2). Select **Port 2** to look into port 1 as input impedance. Click **OK** to close the dialog box. The schematic is analyzed with an auto-selection of frequency range and input impedances are calculated.

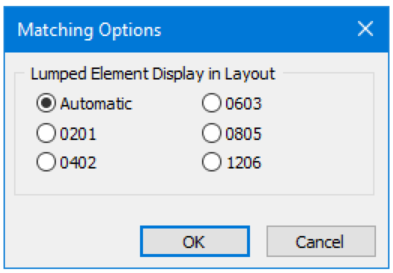
If the methods are successful, the **Freq,R,jX** box is filled with data. Up to 101 rows of data are allowed, the rest are ignored. You can edit the data in the box at another time to trim excess or add more values.

On the right side of the dialog box, source and load impedances are calculated and displayed as Zin1 and Zin2 respectively. The real and imaginary parts are displayed separately. If the terminations are modeled as a combination of R,L,C values, these impedances are exact. If they are given in Freq,R,jX format, the impedances are interpolated at the frequency of interest.

Under **Analysis**, you can enter the minimum and maximum frequencies within which the impedances are calculated and displayed. Click the **Auto Freq Range** check box if you want the wizard to determine those frequencies.

13.8.2.2. Matching Options Dialog Box

The Matching Options dialog box currently contains only a lumped element display choice. For mixed element designs, you must use lumped elements along with transmission lines and/or stubs. Various size options exist for lumped elements. To use a specific element size, select the desired option in the group. If you select **Automatic**, iMatch decides the optimum size of elements based on technology selection and transmission lines widths and lengths in the design. Click **OK** to close the dialog box.



13.8.2.3. Analysis Frequency Range

To change the analysis frequency range quickly, click one of the following toolbar buttons in the middle of the dialog box:

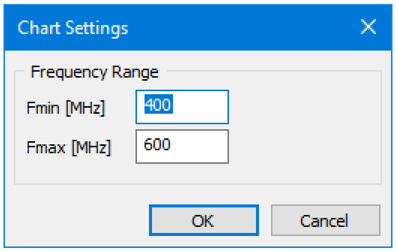
From left to right, these buttons are:

- **Increase Analysis Span**
- **Decrease Analysis Span**
- **Narrow Analysis Span** (center frequency **Fo** is assumed)
- **Wide Analysis Span** (center frequency **Fo** is assumed)
- **Ultra Wide Analysis Span** (center frequency **Fo** is assumed)
- **Auto Span when Passband Changes** (moves the center frequency of the analysis range, along with changes in **Fo**)

13.8.2.4. Chart Setting Dialog Box

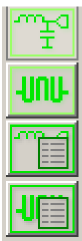
To manually enter the analysis frequency range, click the **Edit Chart Settings** button.

The Chart Settings dialog box displays to allow you to specify **Frequency Range** values.



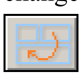
13.8.2.5. Graphics Display Control Options

You can configure the graphics (schematic/layout) side of the dialog box to display schematic, layout, schematic info, or layout info using the following buttons.



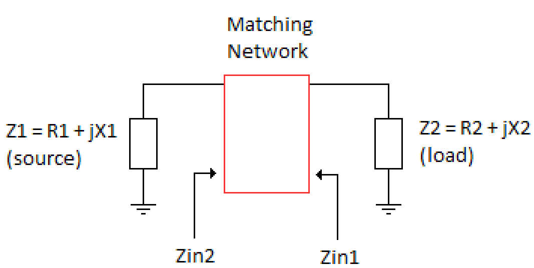
This action is similar to that of the iFilter Wizard. See "Viewing the Schematic and Layout" for more information.

You can rotate the position of the items on the graphics side (schematic/layout, insertion loss-return loss graph and Smith Chart) for better display details by clicking the **Change View Order of Drawing and Plots** button. Note that the windows change places.



13.8.3. Impedance Matching Basics

Impedance matching circuits contain three sections: source termination, matching network, and load termination. This statement also implies a topology which can be shown as a cascaded element schematic, as shown in the following figure.



Conventionally, source and load terminations are shown on the left and right, respectively. In the previous schematic, Z1 (or ZS) is named as the source impedance, and Z2 (or ZL) is named as the load impedance. Zin2 is the input impedance of the load termination after being matched by the matching circuit. Likewise, Zin1 is the input impedance of the source termination after being matched by the matching circuit.

Termination and impedance are used interchangeably in the context of matching and filtering circuits, so source impedance and source termination mean the same thing.

13.8.4. Maximum Power Transfer

Impedance matching is the practice of designing circuits:

- to minimize reflections between source and load terminations, and
- to maximize power transfer from source to load.

Typically, matching circuits contain reactive elements and transmission lines (just like filters), that do not intentionally cause dissipation.

To maximize power transfer, the output impedance of the source termination must be equal to the complex conjugate of the input impedance of the load termination.

In the previous schematic, the following condition provides the maximum power transfer condition:

Z1 = Zin1*

so,

R1 + jX1 = (Rin1 + jX1)* = Rin1 - jXin1

To satisfy this equation,

Rin1 = R1, and

Xin1 = -X1

There are two ways to solve these equations and find a matching network. The first is to perform circuit synthesis by constructing transfer functions for complex terminations and extracting element values. Complex impedance circuit synthesis is cumbersome and very rarely performed in practical designs. The second method is to cancel reactances at the first chance and deal with purely resistive terminations. Many topologies are available with explicit formulations or iterations which can match resistive terminations over satisfactory bandwidths. In more than 95% of applications, the second method is adequate.

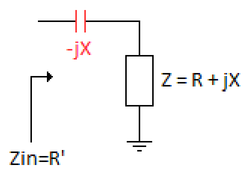
In iMatch, where appropriate, reactive terminations (source or load) are simply turned into resistive networks by applying the cancellation method you choose.

13.8.5. Reactance Cancellation

iMatch provides four reactance cancellation methods in the Matching dialog box **Reactance Cancellation** option. You can select different methods for source and load, as displayed. In the following method descriptions, only the load side is shown. The same methods are applicable to the source side if the source termination is reactive.

13.8.5.1. Lumped (Series) Cancellation Method

In this method, a series IND or CAP is placed next to the termination. If the reactive part of the termination is positive, a negative reactance is needed and a CAP is added. If the reactive part is negative, a positive reactance is needed and an IND is added.



The series element value is calculated from the required reactance and frequency of matching, as specified in the dialog box. For inductors, X = 2*π*Fo*L. For capacitors, X = 1/(2*π*Fo*C).

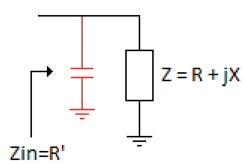
The input impedance seen from the far side of the matching element is now purely resistive and it is the same as the resistive part of the termination, for example, Zin = R’ = R

This method is very simple and effective except for in the following two instances:

- When used in circuits where the series arm is also used for supplying DC currents, a capacitive cancellation is not adequate because a series capacitor is a DC-block.
- Rin is the same as R after cancellation. When R is too small or too large, this may pose a matching problem for intended bandwidths. It may be better to use the shunt cancellation method for extreme values of R.

13.8.5.2. Lumped (Shunt) Cancellation Method

In this method, a shunt IND or CAP is placed next to the termination. If the reactive part of termination is positive, a negative reactance is needed and a CAP is added. If the reactive part of a termination is negative, a positive reactance is needed and an IND is added.



The shunt element value is calculated from the required reactance and frequency of matching, as specified in the dialog box. The required reactance is calculated from Xm = -(R*R+X*X)/X.

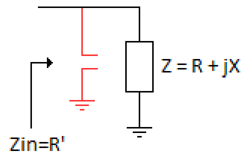
The input impedance seen from the far side of the matching element is now purely resistive and calculated from

Zin = R’ = Xm*Xm*R / (R*R + (Xm+X)*(Xm+X)).

This suggests that the resistive part of input impedance (or simply the input impedance) is no longer equal to R. This offers a big advantage for terminations that have extreme resistive parts. The shunt **Reactance Cancellation** might bring it to reasonable levels. For example, for Z = 1 + j5 Ω, when a shunt -5.2 Ω is added to this termination using a shunt capacitor, the input impedance becomes Zin = 26 Ω. Compared to 1 Ω, 26 Ω offers more matching options and wider bandwidth.

13.8.5.3. Stub (Shunt) Cancellation Method

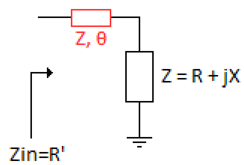
This cancellation method is similar to the lumped (shunt) cancellation method, except the shunt element is an open circuit transmission line stub. Xm and the resultant R’ are calculated the same way. Xm, however, is used to find the length of the stub and the specified stub impedance. You specify stub impedance, Zo, in **Reactance Cancellation** as **Line Imp [ohm]**.



For negative Xm values, the line length is EL = -Zo/Xm. For positive Xm values, 180-degrees is added to EL. Short-circuited stubs are not offered as a solution as they are not practical. In most cases, these circuits are used in matching amplifier inputs and outputs. Short-circuited stubs also short-circuit the gate bias or drain supply to ground, which is not desirable.

13.8.5.4. Transmission Line Cancellation Method

Transmission lines are very useful matching elements, as they can alter the input impedance in a number of ways.



The input impedance of a transmission line terminated by a load impedance ZL is given as:

$$Z_{in} = Z_o \frac{Z_L + jZ_o \tan\theta}{Z_o + jZ_L \tan\theta}$$

By selecting a Zo and Theta, this sophisticated equation can be manipulated to give:

- purely reactive input impedance
- purely resistive input impedance
- a certain VSWR level

In this cancellation method, you specify Zo in **Reactance Cancellation** as **Line Imp [ohm]**. Electrical line length is then calculated to obtain Zin = purely real impedance.

Reactance Cancellation is available in most reactive matching conditions, except for **TL+Stub** types. For **TL+Stub** matching types, an inherent transmission line cancellation is applied as part of the selected matching option.

13.8.5.5. Required Level of Matching

When the impedance matching is “perfect” at a frequency, the circuit has infinite return loss, and the transmission is ideal (lossless). This frequency is also called a “reflection zero” in filter terminology. A reflection zero is rarely achieved in practical circuits due to certain dissipative losses.

You do not need to obtain perfect matching; a level of return loss may be adequate for the application of interest. For example, a 10 to 15dB input return loss may be satisfactory for a power amplifier. The performance difference that is obtained by matching with 20dB return loss is not significant. Otherwise, obtaining a good output match is crucial, as the power of interest is typically 5-10dB higher compared to the input. For example, for a 50W amplifier with a 15dB match on input and output, a 5dB improvement in the match corresponds to 0.2W at the input and 2W at the output. Beyond a 20dB return loss, there is not much gain in terms of transmitted power.

13.8.5.6. Single Frequency Point Matching

Every matching section adds a reflection zero to the return loss, so it improves matching bandwidth. You can add reflection zeros across the bandwidth and obtain wide-band performance. The reflection zeros can also all be gathered at a single frequency to obtain a "deeper" return loss in the center (> 30dB) and shallower return loss at the band corners (10-15dB). The first method requires circuit synthesis with unequal and/or complex terminations. The second method is straightforward and it works well for practical applications. iMatch uses the "single frequency point method", where the matching is performed at a single frequency. Up to 4th order matching circuits are available, which meets most bandwidth requirements even with extreme impedance ratios.

13.8.5.7. Step-by-step or iMatch

In many textbooks, step-by-step impedance matching is described where you can arbitrarily add elements to obtain impedance matching at a single frequency. This method teaches you the physical side of matching in terms of how specific elements contribute to the input impedance of a circuit. This method is worth learning, however there are two drawbacks associated with the step-by-step matching method. First, it takes time to test different matching types and choose the best one based on size, cost, and other variables. Second, it is hard to predict frequency performance, as the matching is only obtained at a single frequency. You can apply techniques such as staying in constant Q circles, but these techniques only give an approximate solution. You still must simulate the circuit in a circuit simulator to determine wideband performance.

In iMatch, you can perform step-by-step matching by clicking the **Manual** button under **Matching**. In manual mode, iMatch uses the manually selected elements and their values and does not perform any further matching. The response and layout (if there is one) are still calculated and drawn as expected.

Other than manual matching, iMatch contains a large library of step-by-step matching combinations. They are also conveniently listed for selection. For example, in a 2-section LC match, you can only use two topologies: lowpass type (series-IND+shunt-CAP) or highpass type (shunt-IND+series-CAP). As these are the only combinations, iMatch makes them both available in selection boxes while presenting schematic, frequency response, and Smith Chart impedance response. With a few mouse clicks you can see the difference between these circuits. The element values are optimally calculated and further changes are not necessary. For this reason, iMatch does not allow you to edit element values. There are enough matching types in the library to allow you to find a suitable solution. Fifty matching circuit types are currently available, so for any given design problem, you can always find a suitable matching type.

13.8.5.8. Smith Chart

The Smith Chart is a graphical representation of transmission line and impedance matching circuits. It is widely used in theoretical work, teaching, and understanding of how various electrical components change the effective input impedance and reflection of high-frequency networks. There are many textbooks and online material available that discuss how to use a Smith Chart.

In iMatch, the Smith Chart displays for completeness purposes. Its main use which exploits "how individual components move an input impedance around the chart at a single frequency" is replaced by the more useful "wideband frequency response". The chart in iMatch shows input/output impedances across the selected frequency range. The following impedance traces always display:

- Load impedance, set by clicking the **Edit Terminations** button
- Complex conjugate of Load impedance
- Source impedance, set by clicking the **Edit Terminations** button
- Complex conjugate of Source impedance
- Matching+Load impedance, the input impedance seen from the input of matching circuit towards the load

Additionally, you can add two optional traces representing the 2nd and 3rd harmonic response of the matching network, by clicking the **Show/Hide Harmonics on Smith Chart** ("H") button.

Each trace displays in a different color. A circle is drawn on one end of the traces to mark the minimum frequency of analysis.

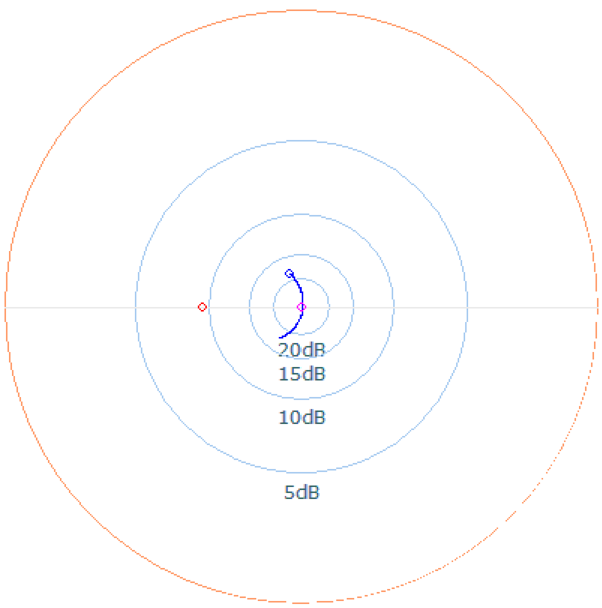
Constant VSWR Circles

Constant VSWR circles are centered on the Smith Chart. As the name implies, at any point along its circumference, VSWR is constant. VSWR and return loss are related by the following equations:

VSWR = (1 + |S11|) / (1 - |S11|)

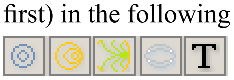
Return Loss = - 10 * log10 |S11|^2

The relationship between VSWR and RL is unique (single-valued). Both terms are used interchangeably in high-frequency circuit design.



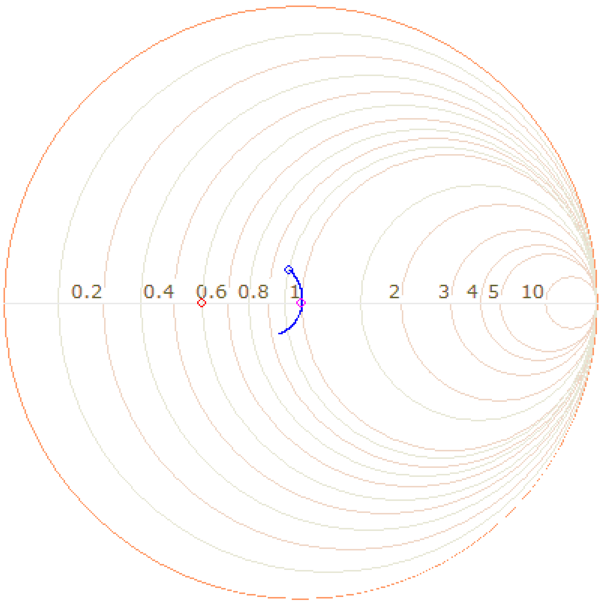
In iMatch, constant VSWR circles that correspond to four major return loss values display. If the impedance stays within the circles, they have better return loss than the circle itself. The aim, therefore, is to contain the impedance trace within the desired constant VSWR circle.

You can toggle constant VSWR circles on a Smith Chart by clicking the **Show/Hide Constant VSWR Circles** button (the first) in the following group of buttons.

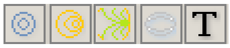


Constant Resistance Circles

Constant resistance circles are centered along the horizontal axis. As the name implies, at any point along its circumference, the resistive part of impedance is constant. The points where the circles cross the horizontal line are “purely” resistive. The circle that passes through the center of a Smith Chart is unity resistance (R=1).

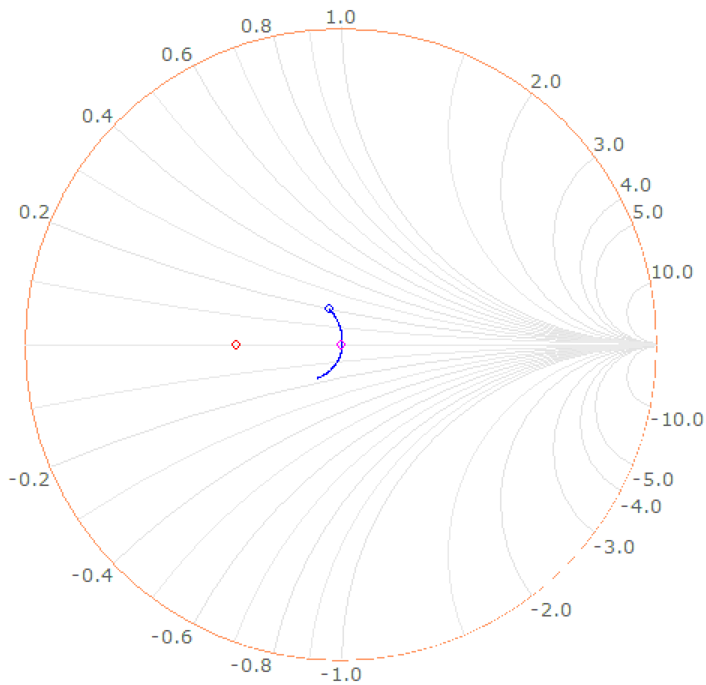


You can toggle constant resistance circles on a Smith Chart by clicking the **Show/Hide Constant Resistance Circles** button (the second) in the following group of buttons.



Constant Reactance Circles

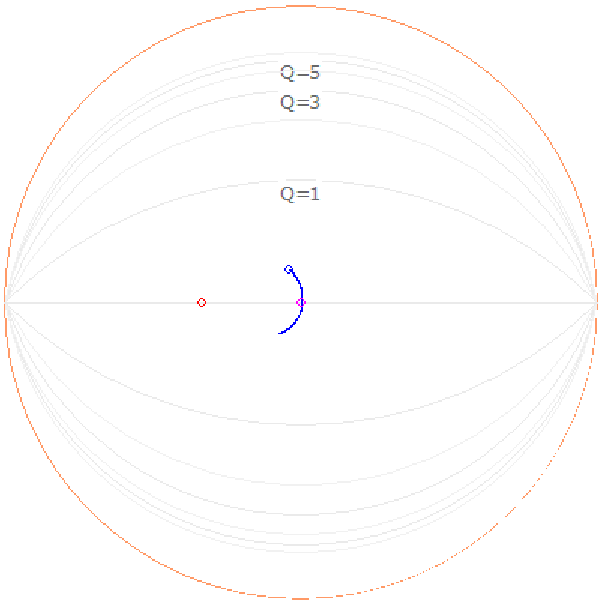
Constant reactance circles are centered along the vertical axis that intersects the right-most point of the horizontal axis on a Smith Chart. As the name implies, at any point along its circumference, the reactive part of the impedance is constant. These circles do not cross the horizontal line which is purely resistive. In the Impedance Chart, the top semicircle is inductive, so the circles represent constant inductance circles. Likewise, the circles in the bottom semicircle are constant capacitance circles.



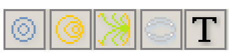
You can toggle constant reactance circles on a Smith Chart by clicking the **Show/Hide Constant Reactance Circles** button (the third) in the following group of buttons.

Constant Q Circles

Constant Q circles are centered along the vertical axis that intersects the center of a Smith Chart. Q is inversely related to the frequency bandwidth of the network. For wideband circuits, it is desirable to stay within the specified constant Q circle. The relationship between Q and bandwidth is not simply interpreted. It is better to use rectangular Insertion Loss/Return Loss charts to understand bandwidth.

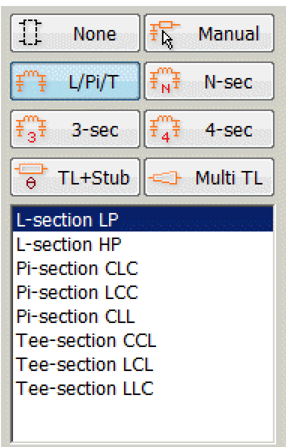


Constant Q circles are displayed for completeness. You can toggle constant Q circles on a Smith Chart by clicking the **Show/Hide Constant Q Circles** button (the fourth) in the following group of buttons.



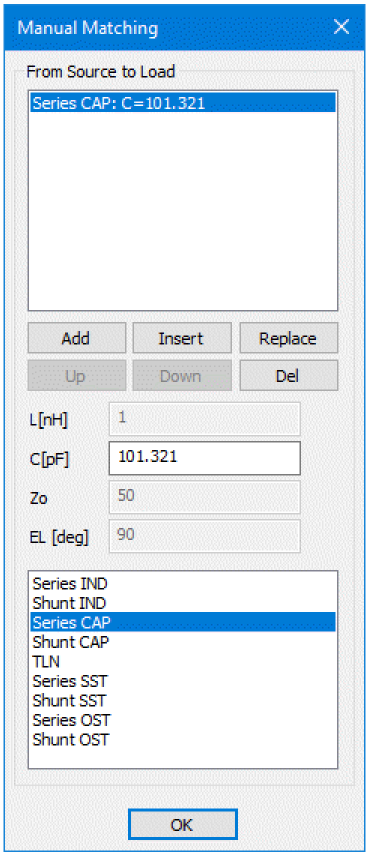
13.8.6. Impedance Matching Types

This section lists impedance matching types currently available in iMatch. To explain matching types, a simple matching example with R1=50, R2=25 is presented. Matching is performed at 500 MHz. For reactive terminations, a reactance cancellation element is included in the matching circuit. Some of the matching types use transmission lines to perform the reactance cancellation, so no extra element is produced.



13.8.6.1. Manual

Manual matching is provided for complementary purposes for those who prefer performing impedance matching at single spot frequencies. In this mode, a Manual Matching dialog box displays to specify matching elements.



At the top of the dialog box, matching elements display in order from source to the load side. This dialog box copies the last matching network when it displays the first time.

You can use the **Add**, **Insert**, **Replace**, and **Del** buttons to modify the matching elements. When an element type is selected in the list box at the bottom of the dialog box, the relevant parameters **L**, **C**, **Zo**, and **EL** of the matching element are editable. The **Up** and **Down** buttons are used to move the position of the selected matching element in the list.

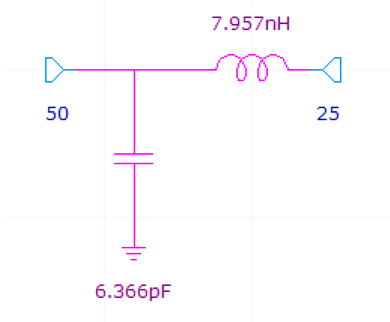
After any changes to the matching network, the main Matching dialog box updates the schematic drawing and the corresponding responses.

13.8.6.2. Lumped Element: L/Pi/Tee Type

These matching types are the simplest 2-element or 3-element sections. Although simple, they provide enough matching for many HF and VHF applications.

L-section LP (Lowpass)

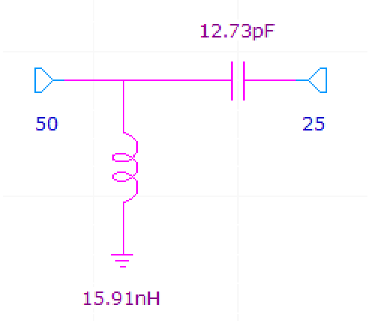
This 2-section lumped element type provides an infinite return loss at the desired matching frequency, and maintains a lowpass frequency response. Due to the Shunt-Series element layout, this circuit is also called "L-section". The position of the series inductor depends on the R1/R2 ratio. If R1 is smaller than R2, then the series inductor is positioned to the left of the shunt capacitor.



This design is a unique solution of impedance values; you can only edit the center frequency. The circuit is DC shorted between the terminations and DC isolated from ground.

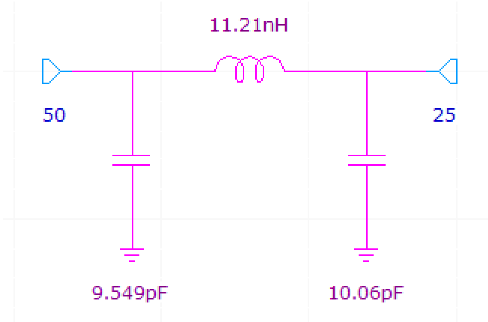
L-section HP (Highpass)

The highpass version of the LP type uses a series capacitor and shunt inductor. The circuit is DC isolated between terminations and DC shorted to ground.



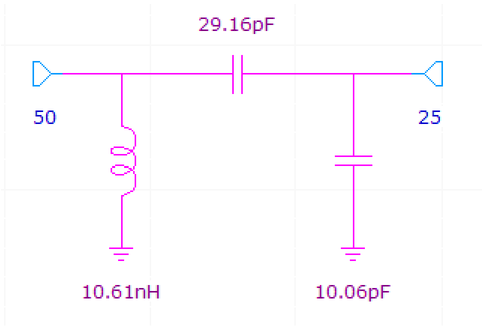
Pi-section CLC (Capacitor–Inductor–Capacitor)

This circuit with two shunt arms and a series element resembles the math symbol π, hence the name. CLC refers to "Capacitor-Inductor-Capacitor". This type of topology is similar to a lowpass filter. The frequency response is lowpass or quasi-lowpass with wideband analysis. The circuit is a DC short between terminations and DC isolated from ground.



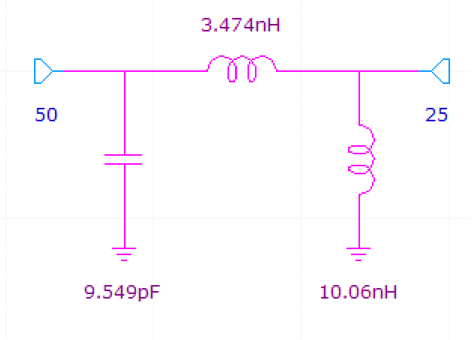
Pi-section LCC (Inductor–Capacitor–Capacitor)

This is another Pi-circuit with a highpass/quasi-highpass/bandpass response. LCC refers to "Inductor-Capacitor-Capacitor". The circuit is DC isolated between terminations and DC shorted to ground.



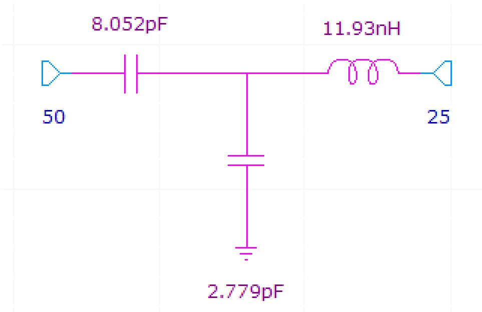
Pi-section CLL (Capacitor–Inductor–Inductor)

This is a Pi-circuit with a moderate bandpass response. CLL refers to "Capacitor-Inductor-Inductor". The circuit is DC shorted between terminations and DC shorted to ground.



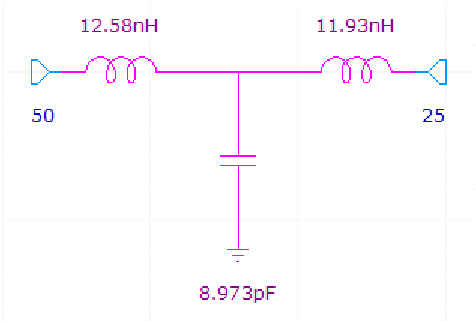
Tee-section CCL (Capacitor–Capacitor–Inductor)

This circuit with two series arms and a shunt element looks like a Tee (letter "T"), hence the name. CCL refers to "Capacitor-Capacitor-Inductor". This type of topology produces a moderate bandpass response, similar to the response of the Pi-section CLL. The circuit is DC isolated between terminations and DC isolated from ground.



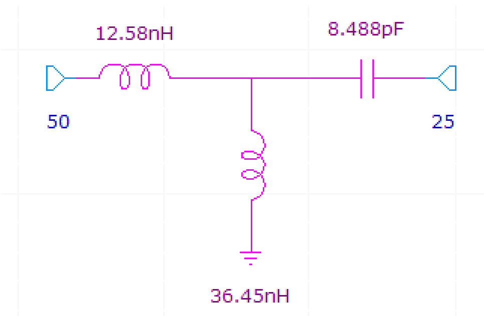
Tee-section LCL (Inductor–Capacitor–Inductor)

This Tee-circuit with a lowpass/quasi-lowpass response is similar to the Pi-section CLC. LCL refers to "Inductor-Capacitor-Inductor". The circuit is DC shorted between terminations and DC isolated from ground.



Tee-section LLC (Inductor–Inductor–Capacitor)

This is a Tee-circuit with a bandpass/quasi-highpass response similar to the Pi-section LCC. LLC refers to "Inductor-Inductor-Capacitor". The circuit is DC isolated between terminations and DC shorted to ground.



13.8.6.3. Lumped Element: N-section

Customized solutions up to 4th order are already provided via dedicated buttons. iMatch also provides generic lowpass type solutions for higher orders. When you select N-section, the parameter **# sections** under **Specifications** is also editable.

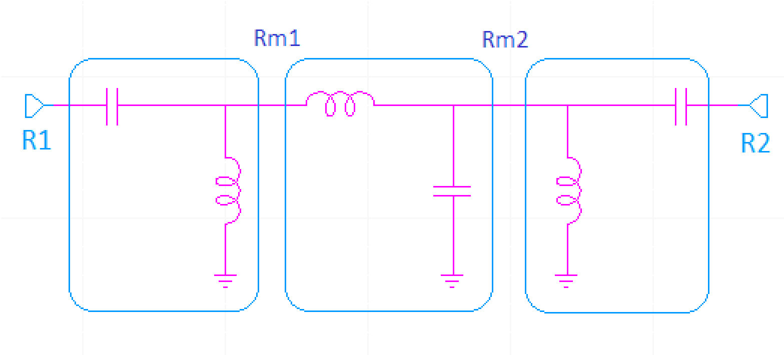
Max.Flat

A Maximally-flat filter provides the flattest passband response at a given frequency. In his September, 1965 MTT article, Cristal provided lowpass prototype tables up to 10th order. iMatch uses those table values and interpolates them for the required impedances.

To calculate the element values, the terminations are first converted to their real-form by performing the selected reactance cancellation method. After obtaining the two real source and load impedances, Z1/Z2 ratio is then looked up in the tables and interpolated for the nearest impedance ratio, and g-values and element values are calculated.

13.8.6.4. Lumped Element: 3-section

3-section lumped element matching circuits are obtained by cascading three lowpass or highpass matching sections. At the frequency of matching, the return loss is very large, so in the vicinity of **Fo**, a bandpass response is obtained. In a wider spectrum, depending on the number of contributing sections, 3-section matching circuits can have either of lowpass, highpass or bandpass responses. The following figure shows a typical matching circuit.



The matching circuit contains HP-LP-HP sections. As the figure suggests, each section is designed to match an impedance level to another one. The first CAP-IND section matches R1 to Rm1, the middle IND-CAP section matches Rm1 to Rm2, and the third IND-CAP section matches Rm2 to R2. Only R1 and R2 are specified by design; you can freely select Rm1 and Rm2. The optimum solution is found when R1/Rm1 = Rm1/Rm2 = Rm2/R2.

You may want to choose different impedance levels to trim the circuit response and adjust element values, however, so at least one of these intermediate values should be left to choice.

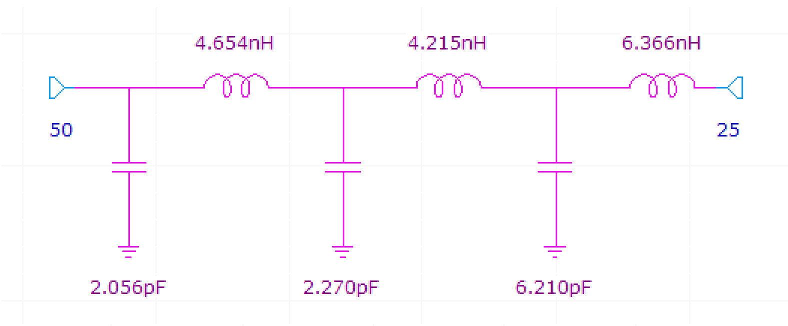
iMatch allows implicit specification of Ra by specifying the Q for the last section. Conventionally, Q gives a better indication of matching bandwidth. Q and Ra are related by the equation Rm2 = R2 * (Q*Q + 1).

After Rm2 is found, Rm1 is calculated from Rm1 = (R1 * Rm2)^0.5 and all three sections are designed using these impedances.

These matching types usually result in six matching elements. In some cases, Q specification yields inner sections that need mirroring. As a result, two parallel or series elements may occur which are combined in the end, and five or less elements may exist in the final design.

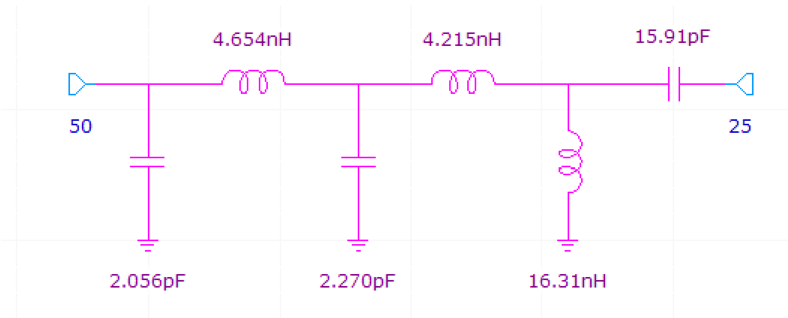
LP-LP-LP (Lowpass–Lowpass–Lowpass)

Due to all three sections having lowpass characteristics, this matching circuit has more lowpass response than bandpass. The circuit is DC shorted between terminations and DC isolated from ground.



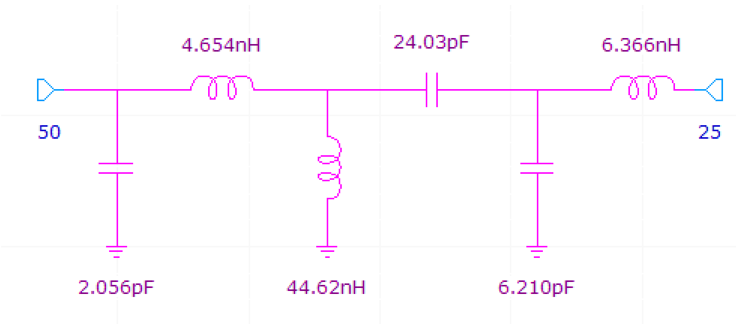
LP-LP-HP (Lowpass–Lowpass–Highpass)

Comprised of lowpass and highpass sections, this matching circuit has a bandpass response. The circuit is DC isolated between terminations and from ground.



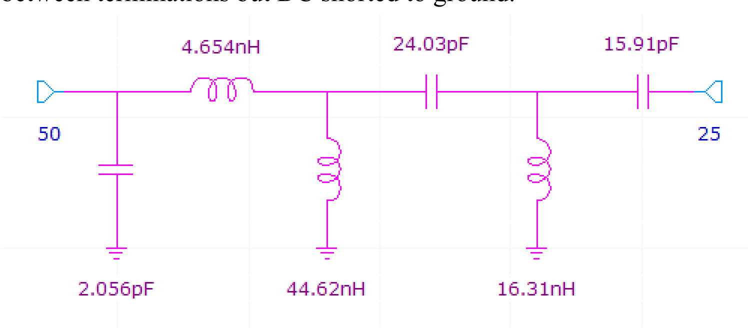
LP-HP-LP (Lowpass-Highpass–Lowpass)

Comprised of lowpass and highpass sections, this matching circuit has a bandpass response. The circuit is DC isolated between terminations but DC shorted to ground.



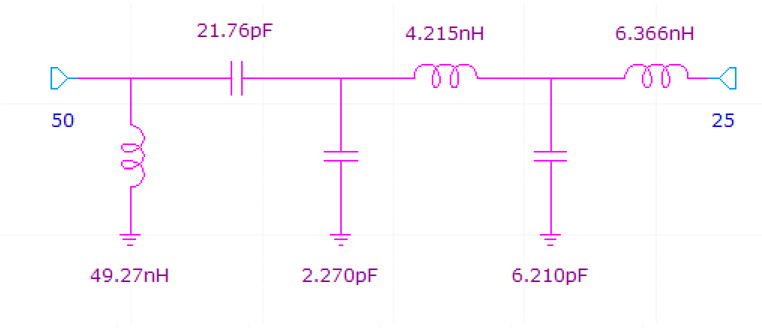
LP-HP-HP (Lowpass–Highpass–Highpass)

Comprised of lowpass and highpass sections, this matching circuit has a bandpass response. The circuit is DC isolated between terminations but DC shorted to ground.



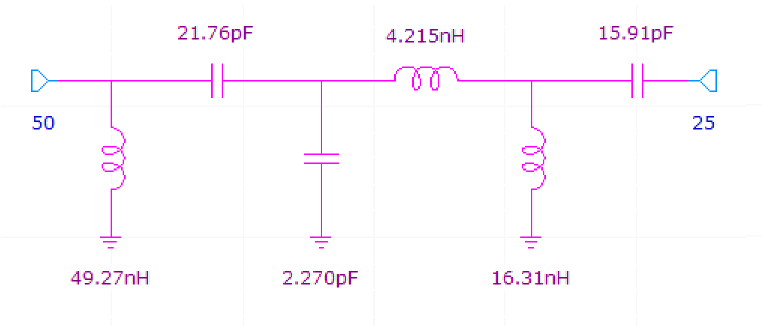
HP-LP-LP (Highpass–Lowpass–Lowpass)

Comprised of lowpass and highpass sections, this matching circuit has a bandpass response. The circuit is DC isolated between terminations but DC shorted to ground.



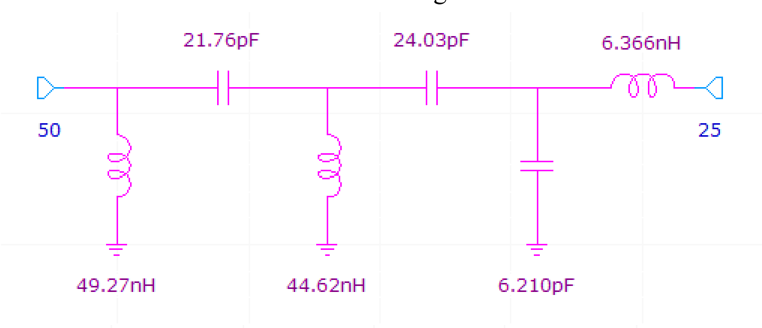
HP-LP-HP (Highpass–Lowpass–Highpass)

Comprised of lowpass and highpass sections, this matching circuit has a bandpass response. The circuit is DC isolated between terminations but DC shorted to ground.



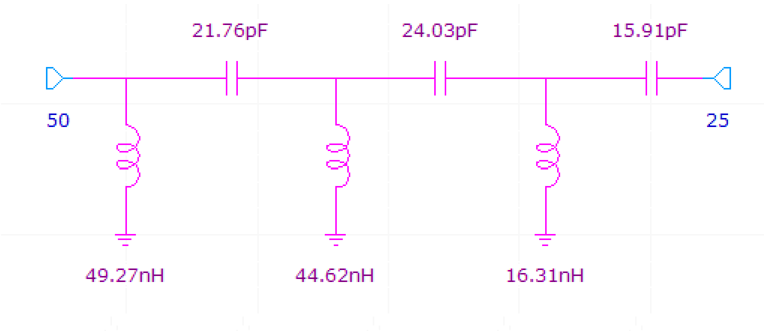
HP-HP-LP (Highpass–Highpass–Lowpass)

Comprised of lowpass and highpass sections, this matching circuit has a bandpass response. The circuit is DC isolated between terminations but DC shorted to ground.



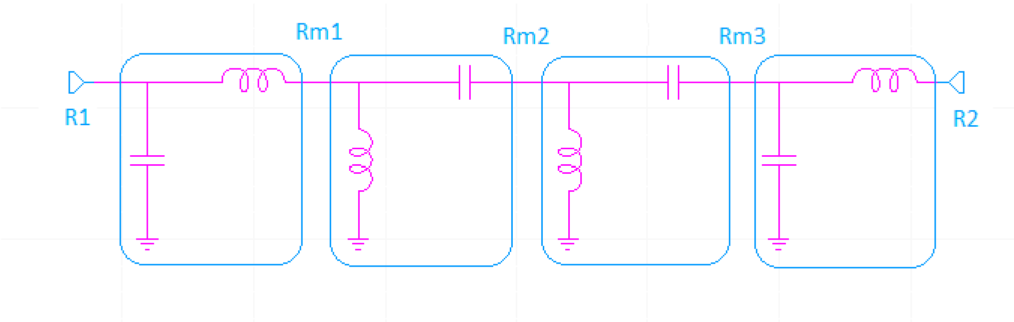
HP-HP-HP (Highpass-Highpass-Highpass)

Comprised of three highpass sections, this matching circuit has more highpass response than bandpass. The circuit is DC isolated between terminations but DC shorted to ground.



13.8.6.5. Lumped Element: 4-section

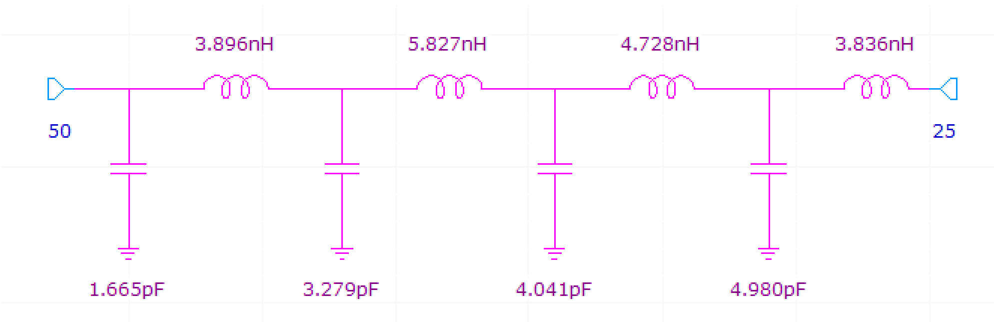
4-section lumped element matching circuits are similar to 3-section circuits, except that they have one more section, and therefore, potentially wider bandwidth. Much of the explanation for 3-section circuits is valid for 4-section circuits.



As in 3-section circuits, Q is specified for the last section which matches Rm3 and R2. This Q specification offers flexibility for impedance levels. Once Rm3 is determined from Q and R2, the other intermediate levels are found with R1/Rm1 = Rm1/Rm2 = Rm2/Rm3.

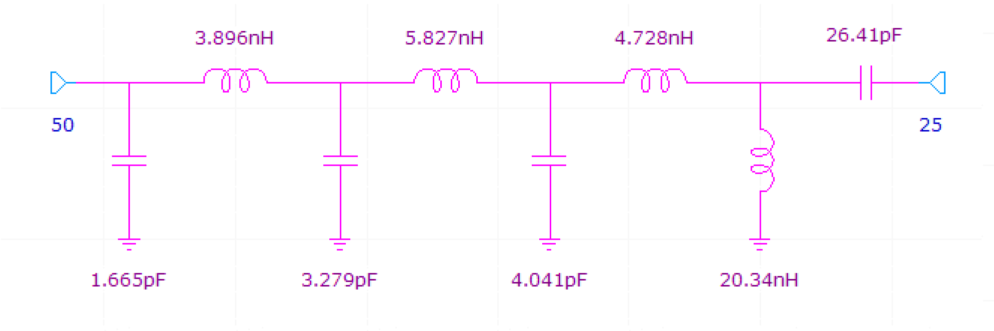
LP-LP-LP-LP (Lowpass–Lowpass–Lowpass–Lowpass)

Comprised of all lowpass sections, this matching circuit has more lowpass response than bandpass. The circuit is DC shorted between terminations but DC isolated from ground.



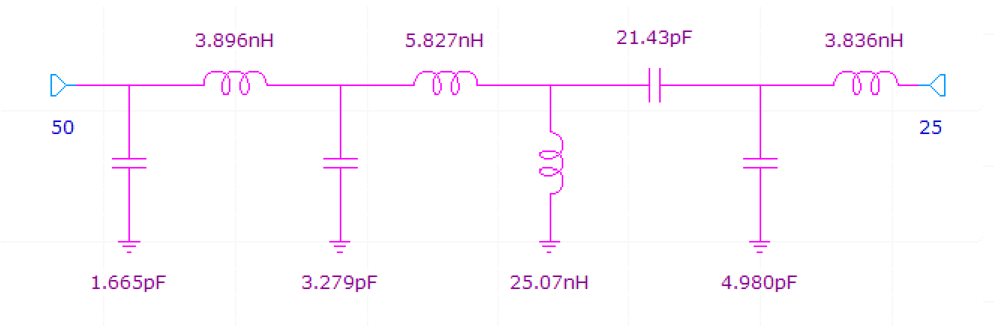
LP-LP-LP-HP (Lowpass–Lowpass–Lowpass–Highpass)

Comprised of lowpass and highpass sections, this matching circuit has a bandpass response. The circuit is DC isolated between terminations but DC shorted to ground.



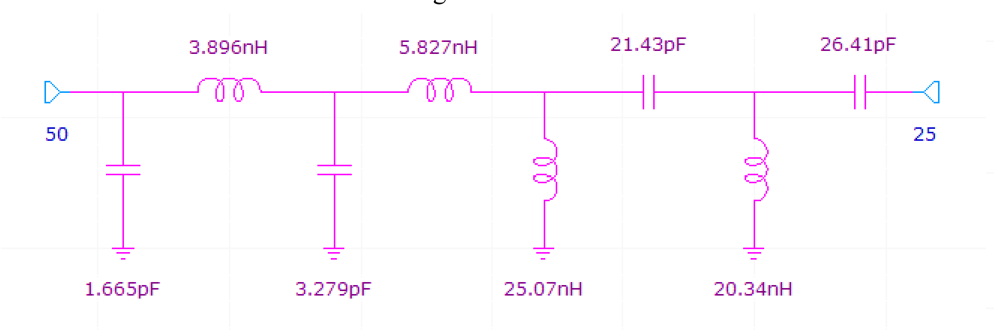
LP-LP-HP-LP (Lowpass–Lowpass–Highpass–Lowpass)

Comprised of lowpass and highpass sections, this matching circuit has a bandpass response. The circuit is DC isolated between terminations but DC shorted to ground.



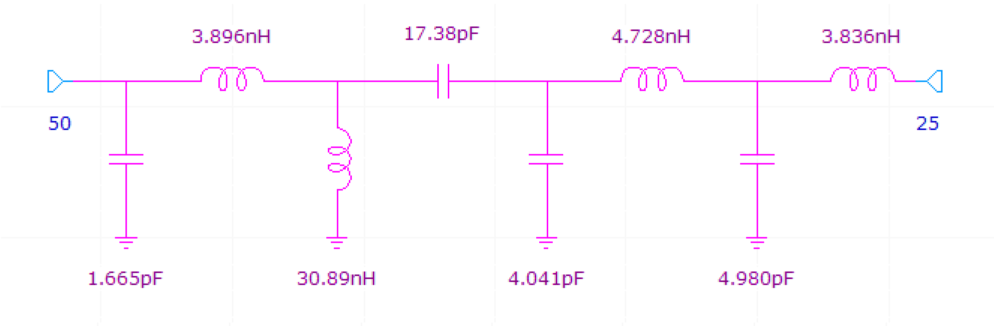
LP-LP-HP-HP (Lowpass–Lowpass–Highpass–Highpass)

Comprised of lowpass and highpass sections, this matching circuit has a bandpass response. The circuit is DC isolated between terminations but DC shorted to ground.



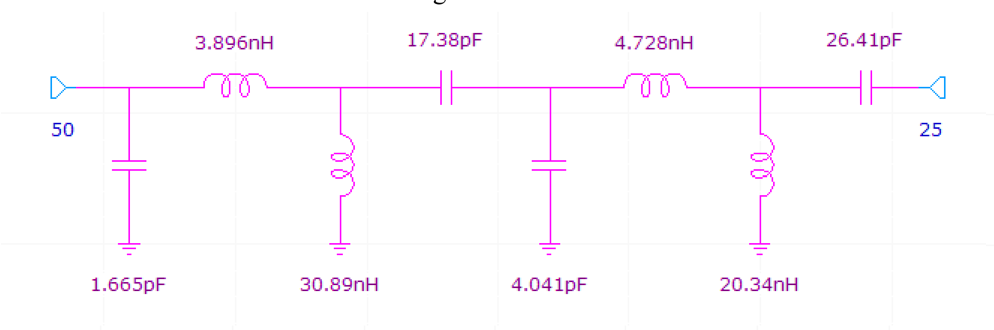
LP-HP-LP-LP (Lowpass–Highpass–Lowpass–Lowpass)

Comprised of lowpass and highpass sections, this matching circuit has a bandpass response. The circuit is DC isolated between terminations but DC shorted to ground.



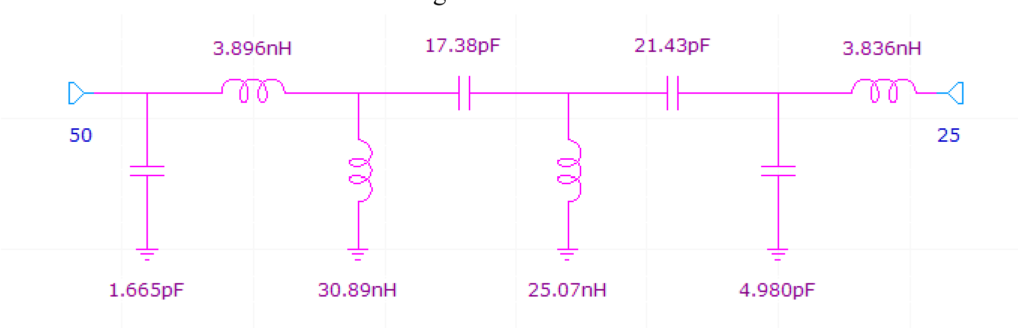
LP-HP-LP-HP (Lowpass–Highpass–Lowpass–Highpass)

Comprised of lowpass and highpass sections, this matching circuit has a bandpass response. The circuit is DC isolated between terminations but DC shorted to ground.



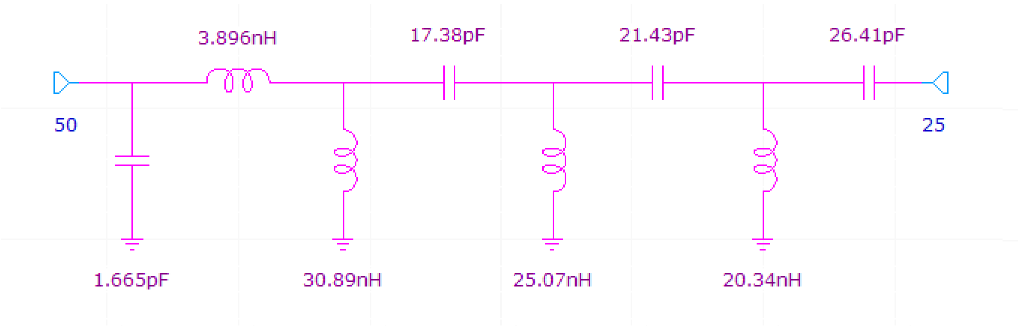
LP-HP-HP-LP (Lowpass–Highpass–Highpass–Lowpass)

Comprised of lowpass and highpass sections, this matching circuit has a bandpass response. The circuit is DC isolated between terminations but DC shorted to ground.



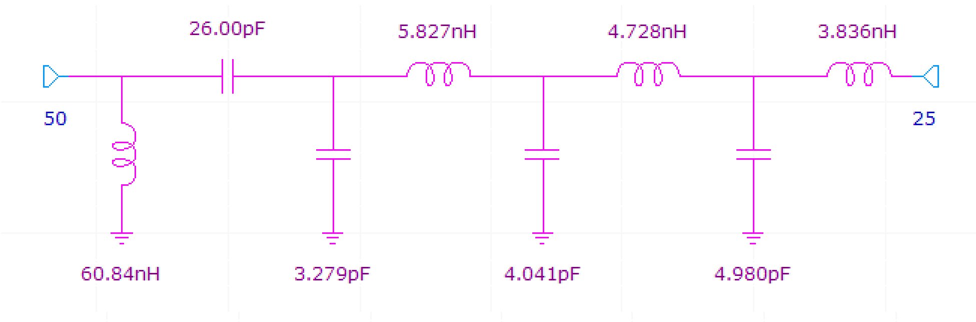
LP-HP-HP-HP (Lowpass–Highpass–Highpass–Highpass)

Comprised of lowpass and highpass sections, this matching circuit has a bandpass response. The circuit is DC isolated between terminations but DC shorted to ground.



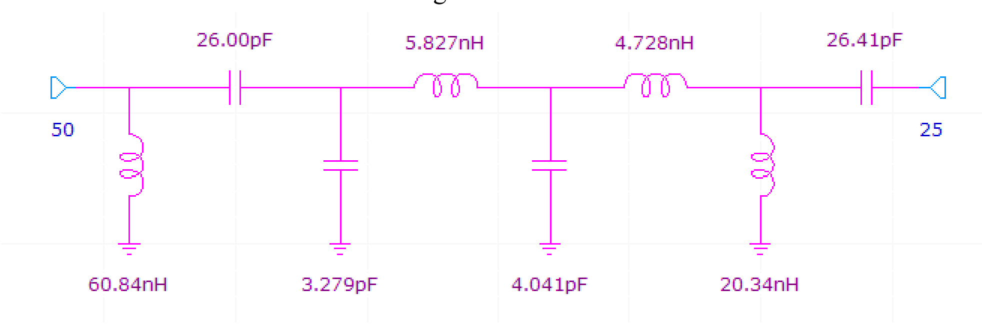
HP-LP-LP-LP (Highpass–Lowpass–Lowpass–Lowpass)

Comprised of lowpass and highpass sections, this matching circuit has a bandpass response. The circuit is DC isolated between terminations but DC shorted to ground.



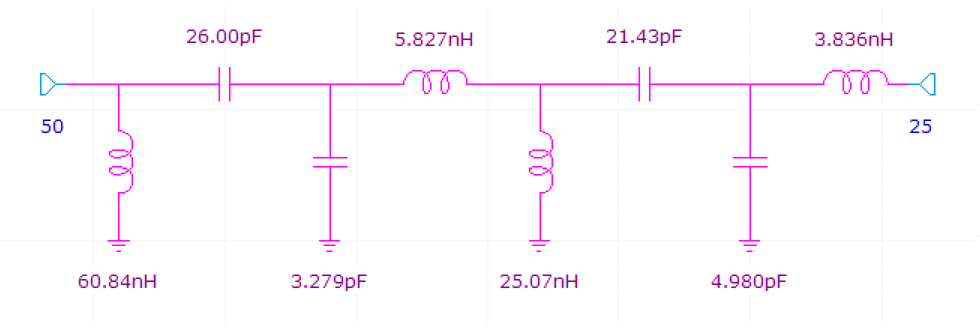
HP-LP-LP-HP (Highpass–Lowpass–Lowpass–Highpass)

Comprised of lowpass and highpass sections, this matching circuit has a bandpass response. The circuit is DC isolated between terminations but DC shorted to ground.



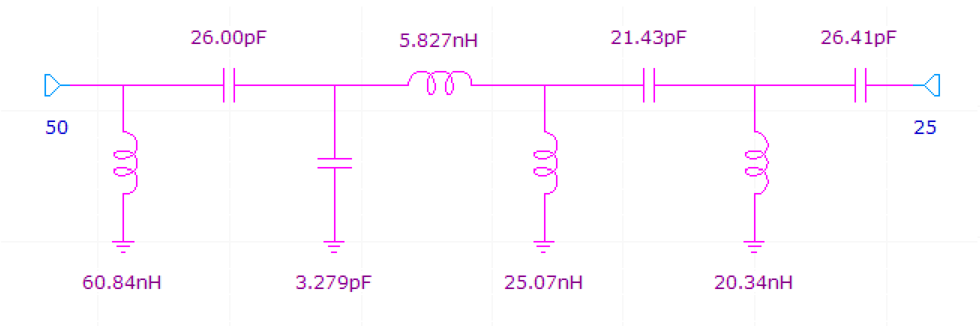
HP-LP-HP-LP (Highpass–Lowpass–Highpass–Lowpass)

Comprised of lowpass and highpass sections, this matching circuit has a bandpass response. The circuit is DC isolated between terminations but DC shorted to ground.



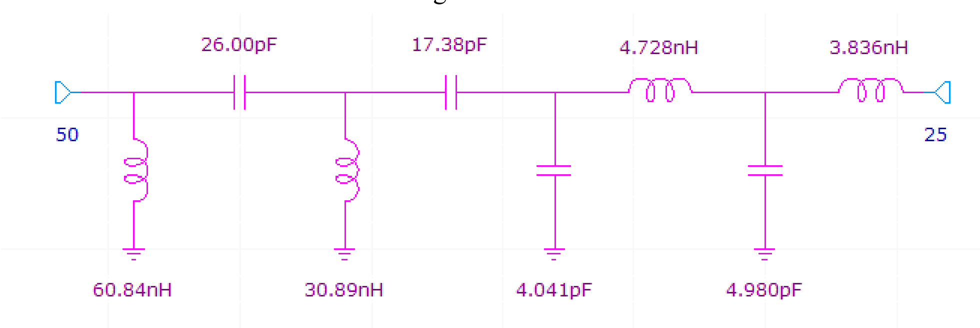
HP-LP-HP-HP (Highpass–Lowpass–Highpass–Highpass)

Comprised of lowpass and highpass sections, this matching circuit has a bandpass response. The circuit is DC isolated between terminations but DC shorted to ground.



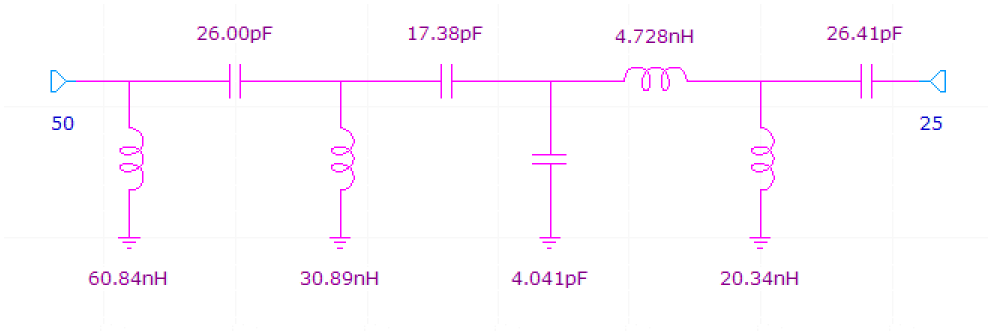
HP-HP-LP-LP (Highpass–Highpass–Lowpass–Lowpass)

Comprised of lowpass and highpass sections, this matching circuit has a bandpass response. The circuit is DC isolated between terminations but DC shorted to ground.



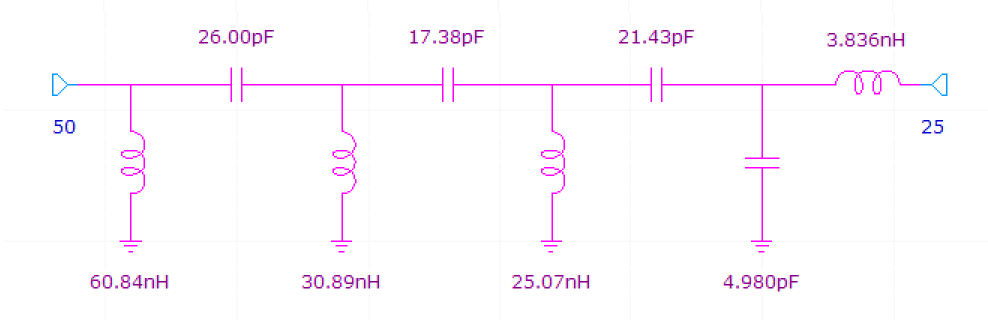
HP-HP-LP-HP (Highpass–Highpass–Lowpass–Highpass)

Comprised of lowpass and highpass sections, this matching circuit has a bandpass response. The circuit is DC isolated between terminations but DC shorted to ground.



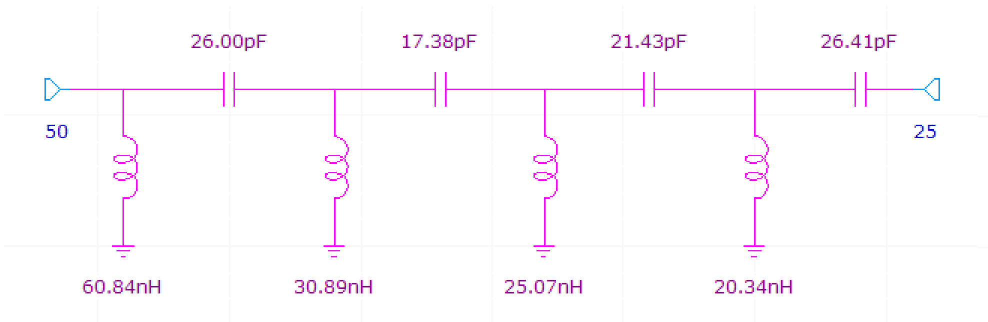
HP-HP-HP-LP (Highpass–Highpass–Highpass–Lowpass)

Comprised of lowpass and highpass sections, this matching circuit has a bandpass response. The circuit is DC isolated between terminations but DC shorted to ground.



HP-HP-HP-HP (Highpass–Highpass–Highpass–Highpass)

Comprised of all highpass sections, this matching circuit has more highpass response than bandpass. The circuit is DC isolated between terminations but DC shorted to ground.



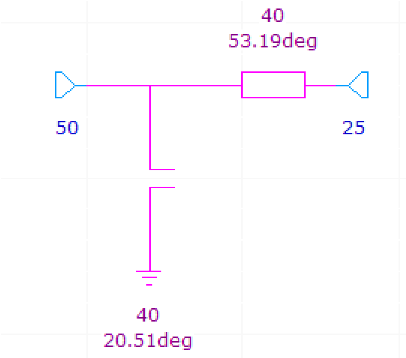
13.8.6.6. Distributed/Mixed Element: TL+Stub

TL+stub matching circuits use distributed or mixed elements to achieve impedance matching at UHF and microwave frequencies. Transmission lines and stubs are mostly printed circuits, requiring you to add only one or two shunt capacitors. Because of their simplicity to construct and tune (trimming lines and stubs), they are by far the most used matching circuits at high frequencies.

The transmission line element near the termination is also used to manipulate and/or cancel reactance, so **Reactance Cancellation** is not needed for these matching types. In iMatch, this option is disabled to avoid any confusion.

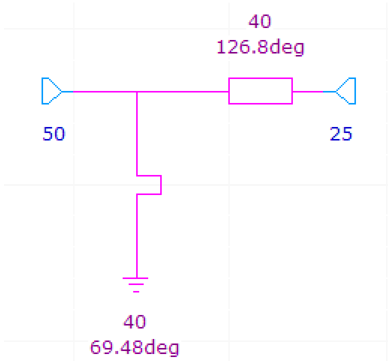
Shunt OST + TL (Shunt Open Stub + Transmission Line)

This distributed element matching network uses a transmission line and an open circuited stub. The same impedance is used for the line and stub, AS specified in **Reactance Cancellation**. The circuit is DC shorted between impedances. Because it is DC isolated from ground, it is suitable for amplifier input and output matching.



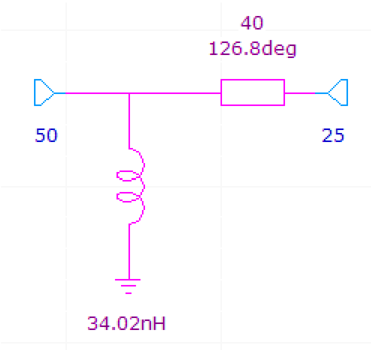
Shunt SST + TL (Shunt Shorted Stub + Transmission Line)

This distributed element matching network uses a transmission line and a short circuited stub. The same impedance is used for the line and stub, as specified in the **Reactance Cancellation** group. The circuit is DC shorted between terminations. Because it is DC shorted to ground, it is not suitable for amplifier input and output matching.



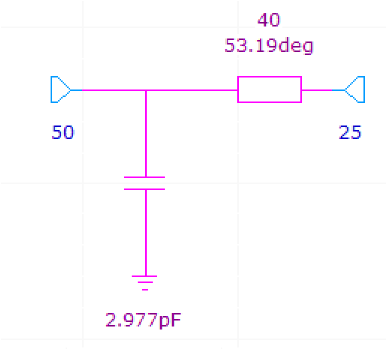
Shunt IND + TL (Shunt Inductor + Transmission Line)

This mixed element matching network uses a transmission line and a shunt inductor. The transmission line impedance is specified in **Reactance Cancellation**. The circuit is DC shorted between terminations. Because it is DC shorted to ground, it is not suitable for amplifier input and output matching.



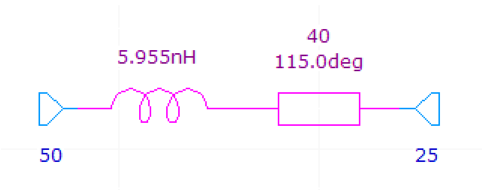
Shunt CAP + TL (Shunt Capacitor + Transmission Line)

This mixed element matching network uses a transmission line and a shunt capacitor. The transmission line impedance is specified in **Reactance Cancellation**. The circuit is DC shorted between terminations. Because it is DC isolated from ground, it is suitable for amplifier input and output matching.



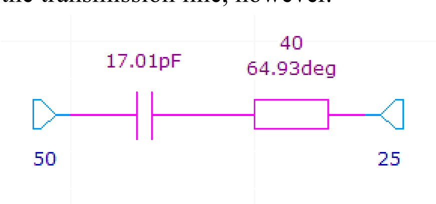
Series IND + TL (Series Inductor + Transmission Line)

This mixed element matching network uses a transmission line and a series inductor. The transmission line impedance is specified in **Reactance Cancellation**. The circuit is DC shorted between terminations. Because it is DC isolated from ground, it is suitable for amplifier input and output matching.



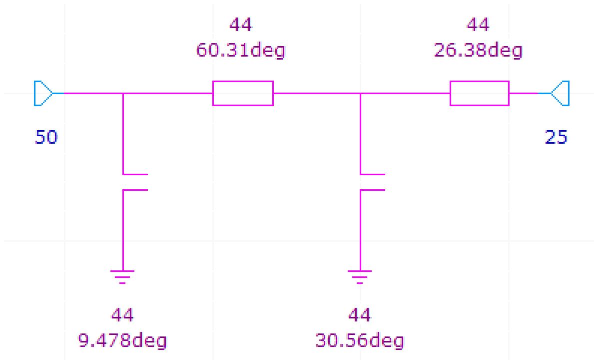
Series CAP + TL (Series Capacitor + Transmission Line)

This mixed element matching network uses a transmission line and a series capacitor. The transmission line impedance is specified in **Reactance Cancellation**. The circuit is DC isolated between impedances. Because it is DC isolated from ground, it is suitable for amplifier input and output matching. The high impedance DC bias line should be connected to the transmission line, however.



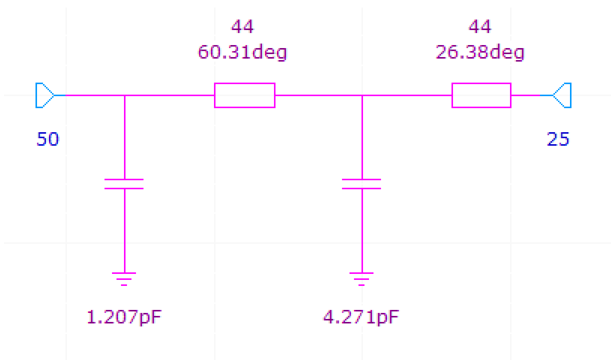
Double Shunt OST + TL (Shunt Open Stub + Transmission Line + Shunt Open Stub + Transmission Line)

This distributed element matching network is obtained by applying Shunt OST + TL twice. An intermediate impedance level Rm = (R1*R2)^0.5 is assumed and the two sections are designed to match R1 to Rm and Rm to R2. The transmission line and stub impedances are all the same and are specified in **Reactance Cancellation**. The circuit is DC shorted between terminations. Because it is DC isolated from ground, it is suitable for amplifier input and output matching.



Double Shunt CAP + TL (Shunt Capacitor + Transmission Line + Shunt Capacitor + Transmission Line)

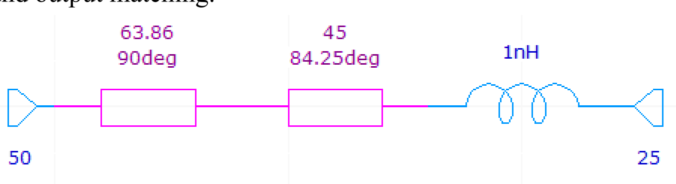
This distributed element matching network is obtained by applying Shunt CAP + TL twice. An intermediate impedance level Rm = (R1*R2)^0.5 is assumed and the two sections are designed to match R1 to Rm and Rm to R2. The transmission line impedances are the same and the value is specified in **Reactance Cancellation**. The circuit is DC shorted between terminations. Because it is DC isolated from ground, it is suitable for amplifier input and output matching.



Double TL (Transmission Line + Transmission Line)

This distributed element matching network is obtained by cascading two transmission lines. You specify the first transmission line impedance near the termination in **Reactance Cancellation**. Its line length is calculated to convert a reactive termination to a real impedance (Rm). In the previous example, an inductor is added to the termination to make it reactive. If the termination is purely resistive, this transmission line is simply omitted. The next transmission line is a quarterwave length line and its impedance is calculated from Zo = (R1*Rm)^0.5.

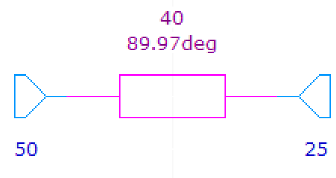
The circuit is DC shorted between terminations. Because it is DC isolated from ground, it is suitable for amplifier input and output matching.



Single TL (short) (Single Transmission Line – Short Line)

Of all the matching networks available, the single transmission line is the only type that does not offer a perfect match at the specified frequency. Its inclusion is only due to its simplicity, which may be preferred for “good-enough” return loss. Given the characteristic impedance, the line length is calculated for the best return loss at the specified frequency. Because of periodicity in a distributed element circuit, line length can be increased in 180-degree increments with the same response. If the line length is too small, you may prefer the Single TL (long) solution which exploits this idea.

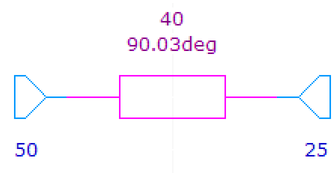
The circuit is DC shorted between terminations. Because it is DC isolated from ground, it is suitable for amplifier input and output matching.



Single TL (long) (Single Transmission Line - Long Line)

Of all the matching networks available, the single transmission line is the only type that does not offer a perfect match at the specified frequency. Its inclusion is only due to its simplicity, which may be preferred for “good-enough” return loss. Given the characteristic impedance, the line length is calculated for the best return loss at the specified frequency.

The circuit is DC shorted between terminations. Because it is DC isolated from ground, it is suitable for amplifier input and output matching.



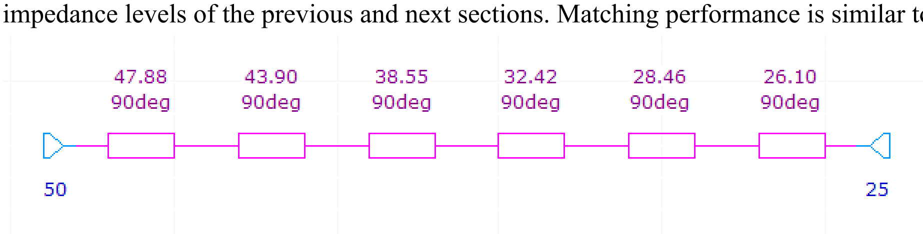
13.8.6.7. Distributed Element: Multiple TL

Multiple transmission line matching circuits are used to obtain wideband matching for large impedance ratios. In addition to the application frequency, you also specify the number of sections. Higher sections result in wider bandwidth but larger circuits. Types in this category yield similar responses with subtle differences as explained for each type. All of the types are DC shorted between terminations and DC isolated from ground, so suitable for amplifier input and output matching.

Multi TL matching circuits do not inherently cancel the reactive parts of terminations, so **Reactance Cancellation** options are available and utilized if the terminations are reactive.

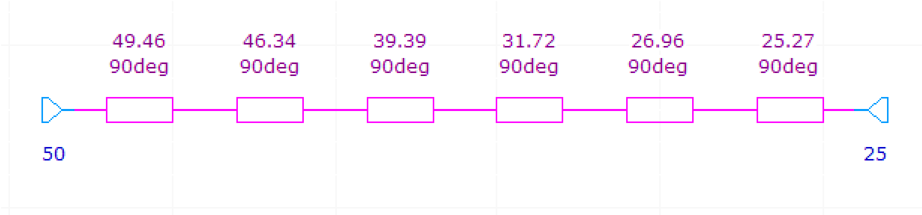
Middle Impedance

This matching network uses the middle impedance method, where intermediate impedance levels and line impedances are calculated from application of the same formula Rm = (Rj*Rk)^0.5, where Rj and Rk belong to impedances or impedance levels of the previous and next sections. Matching performance is similar to the Binomial type.



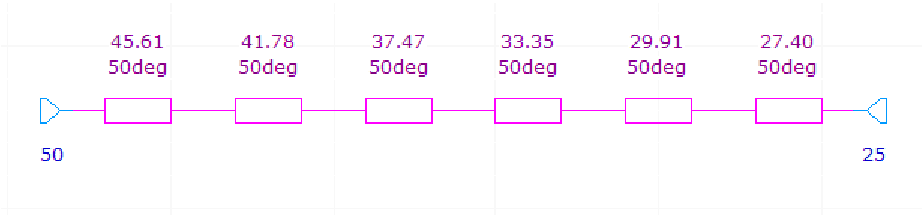
Binomial

This matching network uses the middle impedance method, where intermediate impedance levels and line impedances are calculated from a binomial formula, which gives maximally flat response for 90-degree line lengths. Matching performance is similar to the Middle Impedance type.



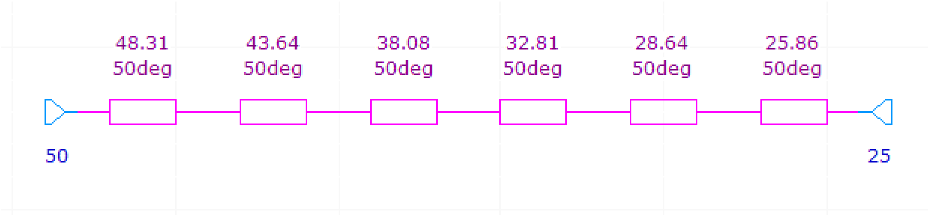
Klopfenstein Taper

Among multi TL types, the Klopfenstein taper provides the widest matching bandwidth for a given total line length. You specify the total line length (EL). Shorter EL causes wider bandwidth, however the lower cutoff frequency of matching increases by decreasing EL. The return loss is uniform across the bandwidth. Klopfenstein tapers can be designed for any return loss.



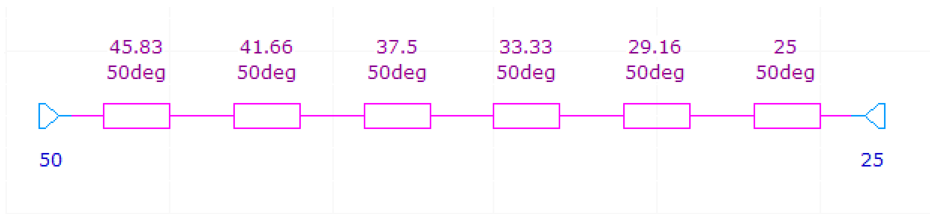
Hecken Taper

The Hecken taper is specified and designed similar to Klopfenstein tapers, but provides more return loss around the matching frequency and less return loss away from the matching frequency.



Exponential Taper

The exponential taper is included for completeness. The impedance values are calculated to fit an exponential increase and the line length is obtained by dividing the specified EL into the number of sections. The matching performance is not significantly better or worse than other multi TL types.



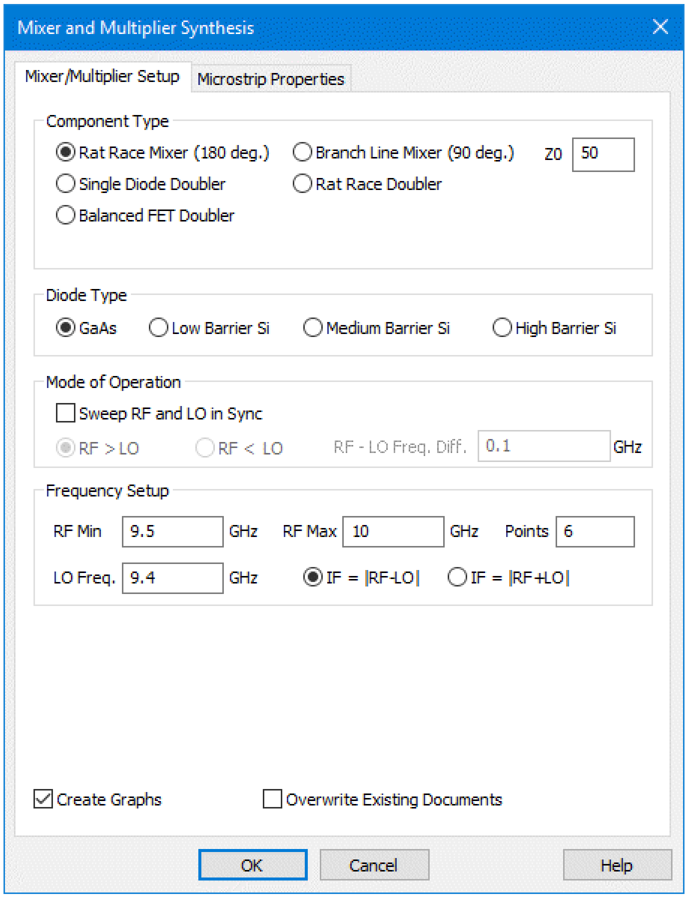
13.9. Mixer and Multiplier Synthesis Wizard

The Mixer and Multiplier Synthesis Wizard allows you to synthesize several types of mixer and multiplier structures to be implemented in microstrip transmission line structures:

- Singly balanced **Rat Race Mixer (180 deg.)** and **Branch Line Mixer (90 deg.)** - These components create similar projects, differing primarily in the type of hybrid used in the circuit. The most important subcircuits are called "MxrElement1" and "MxrElement2". These include stubs that have a tuning function as well as bypassing the diode at appropriate frequencies; they can be tweaked to optimize the passband shape and return loss. The hybrid's center frequency can be adjusted via the Scale variable.
- Simple **Single Diode Doubler** - This component creates a single-diode resistive frequency doubler using a Schottky diode (not a varactor). The doubler usually has approximately 10 dB conversion loss, and output power depends on the capabilities of the diode. A shorted stub on the input side of the diode realizes the second-harmonic current return, and a stub on the output creates a current return for the fundamental frequency. The latter stub is one-quarter wavelength long at the output frequency, making it one-eighth wavelength at the fundamental. As such, it is not a perfect return, and its length can be adjusted to trade off second-harmonic rejection against efficiency. The input stub can also be adjusted to optimize input return loss and to center the passband.
- Balanced **Rat Race Doubler** - This component creates a two-diode, balanced frequency doubler. The circuit is superior to the single-diode circuit in its power handling and even-harmonic rejection, but it is a larger circuit, requiring a rat-race hybrid. The project creates graphs of the hybrid's performance as well as the complete doubler.

  The input stub of the DblElement subcircuit can be used for tuning. The element has no output stub, as the junction between the two diodes is a virtual ground. The lack of an output stub allows greater bandwidth than the single-diode mixer and it allows a lower-impedance return for the fundamental-frequency diode currents.
- **Z0** - Specify the component's port impedance. 50 ohms is the default.

To access the Mixer and Multiplier Synthesis Wizard, open the **Wizards** node in the Project Browser and double-click **Mixer and Multiplier Synthesis**. The Mixer and Multiplier Synthesis dialog box displays.



On the **Mixer/Multiplier Setup** tab, select a **Component Type** and then specify the design parameters:

- **Diode Type** - You can specify any of four diode types: **GaAs** and low- medium- or high-barrier silicon. The synthesis inserts a diode having typical parameters for the diode type and the frequency of operation. You must then select an available diode having similar parameters and substitute the wizard-selected parameters with those of the desired diode. The silicon-diode parameters are based on typical, commercially-available beam-lead devices, and the GaAs diode is based on a MMIC element. The parameters of diodes used in these circuits are usually not critical. In selecting a diode to replace the one inserted by the wizard, you should view the junction capacitance as the most important parameter to be matched.
- **Mode of Operation** - This option allows you to define a wide range of mixer frequency plans: upconverters or downconverters, low- or high-LO mixers, fixed IF or LO frequencies. When the **Sweep RF and LO in Sync** check box is selected, a project is created in which the RF and LO signals are both swept and have a fixed difference frequency. This is most useful in ordinary downconverters having a fixed IF frequency. When selected, you can specify either a high- or low-side LO and the RF-LO difference frequency.
- **Frequency Setup** - If **Sweep RF and LO in Sync** is selected, the frequency setup involves specifying only the RF frequency range in **RF Min** and **RF Max** and the number of **Points** in the swept range. If **Sweep RF and LO in Sync** is cleared, the wizard creates a project in which the RF frequency is swept and the LO frequency is fixed, and values are entered in the boxes for all quantities. You also specify whether the IF is the RF-LO difference (a conventional downconverter) or sum (upconverter). The LO can be either above or below the RF range. If a harmonic mixer is desired, you should enter the fundamental LO frequency. You can then manually modify the measurements in the project to display the desired mixing product. Even for an upconverter, the IF is always the output frequency and the RF is the input. The LO is always the large-signal excitation. Default values of the number of harmonics for the RF and LO, and other

harmonic-balance setup parameters, are created by the wizard. You should review these values to ensure they are sensible for the particular design and its application.

- **Create Graph** or **Overwrite Existing Documents** - Select an option to create a new graph or overwrite an existing graph.

Click **OK** to synthesize the component and send the design to the Microwave Office program. The wizard creates a large number of subcircuits and output graphs. Each mixer or multiplier circuit is hierarchical, having separate subcircuits for the hybrid, diodes and matching circuits, and test circuits for fixed and swept LO power. Several graphs document the performance of each circuit.

On the **Microstrip Setup** tab you can configure the microstrip technology parameters for the components:

- **Substrates** - Select a substrate type with its editable default parameters, or select **Global Definitions MSUB**, a substrate that is already defined in the project's Global Definitions and available from the drop-down list.
- Substrate parameters - You can modify the default parameters for the selected **Substrate**.
- **Length Units** - Select the desired length units. It is always best if the units are the same as the project units, as this is the most precise. In any case, the synthesized component's dimension are entered in project units when the Microwave Office project is created.
- **Tee/Step Type** - Specify discontinuity elements that use either the **Closed Form** or electromagnetic (**EM**) models.

The synthesized project includes the Microwave Office schematic, necessary subcircuits, and graphs, but not finished layouts; you need to manipulate the layout to finalize the component design. Appropriate quantities in the subcircuit are entered as tunable variables. Line lengths in the synthesized circuit are modified to compensate for the size of interconnects and bends. That compensation may not be exact in all cases, causing the center frequency to differ from the value entered. To compensate, the synthesized circuit includes a "scale" variable, which proportionately scales the lengths of all relevant microstrip elements, allowing adjustment of the center frequency.

13.10. Network Synthesis Wizard

The Cadence Network Synthesis networks synthesis wizard is a tool for creating optimized two-port matching networks composed of discrete and distributed components. You specify the maximum number of sections and the types of components to include in the search space. The wizard searches for the best circuit topologies and optimizes the component parameter values.

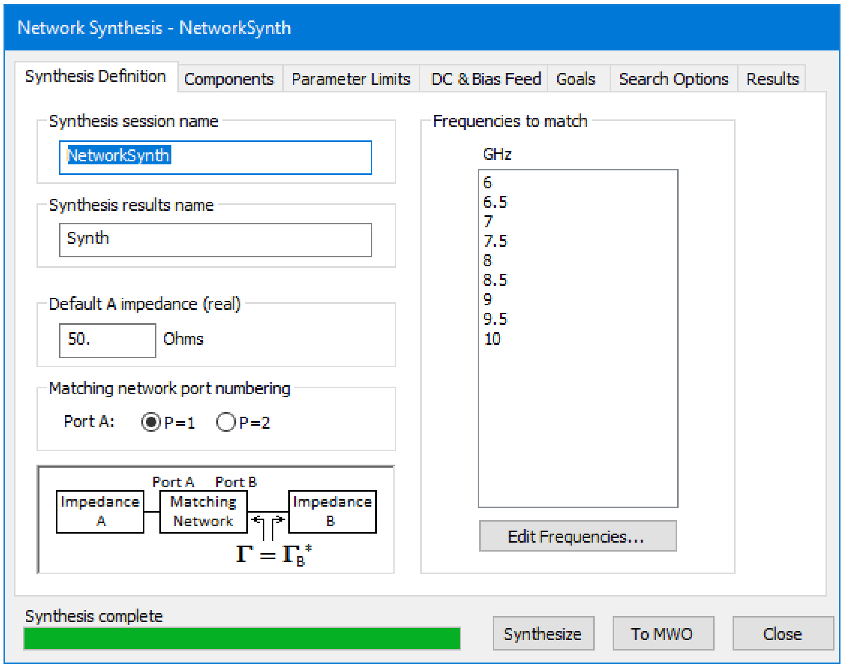
The optimization goals are specified in the wizard using a dedicated set of synthesis measurements, much like optimization goals are normally defined in the AWR Design Environment software. Specialized measurements are provided for input noise matching, amplifier output power matching, and interstage matching. The optimum reflection coefficients are specified over frequency and can be provided in the form of load pull data, network parameter data files, or circuit schematics.

Options allow you to specify DC constraints on the networks and optimally attach a user-provided bias-feed network to the circuit. Also, the component parameter values may be confined within minimum and maximum limits and additionally may be constrained to discrete values.

For information on using the wizard via the AWR Design Environment platform API, see "Network Synthesis Wizard".

13.10.1. Synthesis Definition Tab

At the bottom left of the **Synthesis Definition** tab a block diagram defines the terminology used in the wizard. The rectangle in the middle of the diagram represents the network that the wizard will synthesize. Attached to Port A of that network is a block that represents impedance A. That impedance might be the source impedance of an LNA or the output load for a PA. Connected to Port B of the matching network is impedance B, which for an LNA is the input impedance of the active device and for a PA is the output impedance of the active device. The wizard creates a network to optimize the match between Port B and the block labeled Impedance B.



After the wizard synthesizes the matching networks, the selected networks are drawn in project schematics when you click the **To MWO** button. By default, Port A in the network is port "1" and Port B is port "2". You can reverse this by selecting **P=2** as the **Matching network port numbering**.

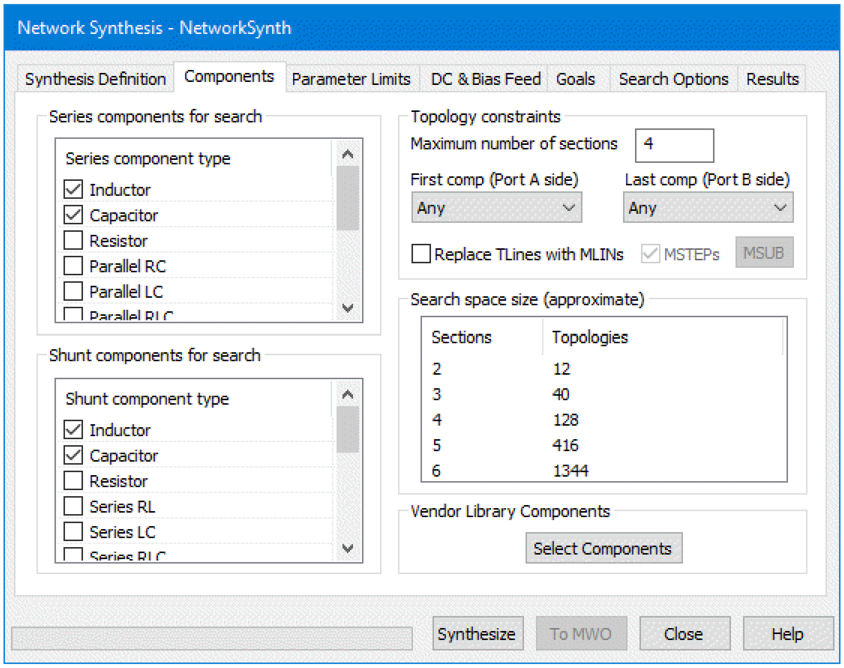
Some of the synthesis measurements allow you to specify impedance A as a parameter for the measurement. If this is not specified, impedance A is taken from **Default A impedance**.

This tab includes two name fields. **Synthesis session name** specifies the name for the wizard saved state, listed under the **Network Synthesis** wizard node in the Project Browser. **Synthesis results name** is used as a base name for the generated networks and schematics and for the user folder in which the schematics are saved.

The **Frequencies to match** list specifies where the measurements are made for calculating the cost of a network during optimization. This list is initialized with the project frequencies. Click **Edit Frequencies** to customize the list.

13.10.2. Components Tab

The **Components** tab provides options for tailoring the set of network topologies over which the wizard searches.



Select the desired components in the shunt and series component lists. The components at the top of the lists are ideal, lumped components, followed by TLines, which are ideal transmission lines. The wizard replaces the TLines with microstrip lines in an additional optimization step if **Replace TLines with MLINs** is selected. Microstrip tees are also added where needed, and microstrip step junctions are added if **MSTEPs** is selected.

By default when MLINs are used, the wizard gets the substrate definition from the first MSUB element it finds when looking through the Global Definitions windows in the project. If there are multiple MSUBs in the project, you should click the **MSUB** button to specify the correct substrate definition.

At the end of the component lists are the vendor library component categories: **Vendor Lib Inductor**, **Vendor Lib Capacitor**, and **Vendor Lib Resistor**. Select these to have the wizard choose between non-ideal components available in vendor component libraries. (The wizard does not vary the values of parameters on these components–if the components have parameters, the default values are used.) Click the **Select Components** button to display the "Select Vendor Library Components for Network Synthesis Dialog Box" where you can edit the lists of allowed library components.

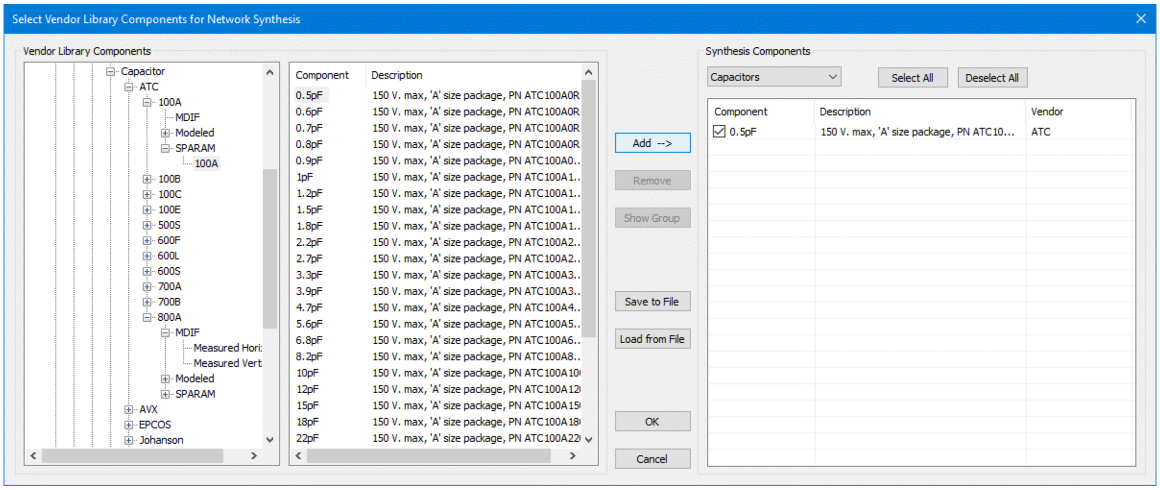
Valid topologies are determined by the types of components selected and the value specified for **Maximum number of sections**. Each section is either a series component or a shunt component. The wizard considers topologies having the maximum number of sections "N", and with fewer, down to N-3 sections. In its synthesis algorithm, any number of series components may be connected in series, but shunt components must have at least one series component between them.

In a typical RF/microwave circuit, some minimum length of transmission line is needed between the active device and any components used in a matching network. The **First comp (Port A side)** and **Last comp (Port B side)** drop-down lists provide a way to specify that a series TLine is required. Alternatively, the first and/or last components can be forced to any other type of component. On the **Parameter Limits** tab, you can specify separate limits for the parameters of these first/last components.

The **Search space size** display is for informational purposes only. It provides an indication of how the size of the search space (number of circuit topologies) increases as a function of the number of sections in the networks and the selection of series and shunt component types.

13.10.2.1. Select Vendor Library Components for Network Synthesis Dialog Box

To access this dialog box, click the **Select Components** button on the **Components** tab.



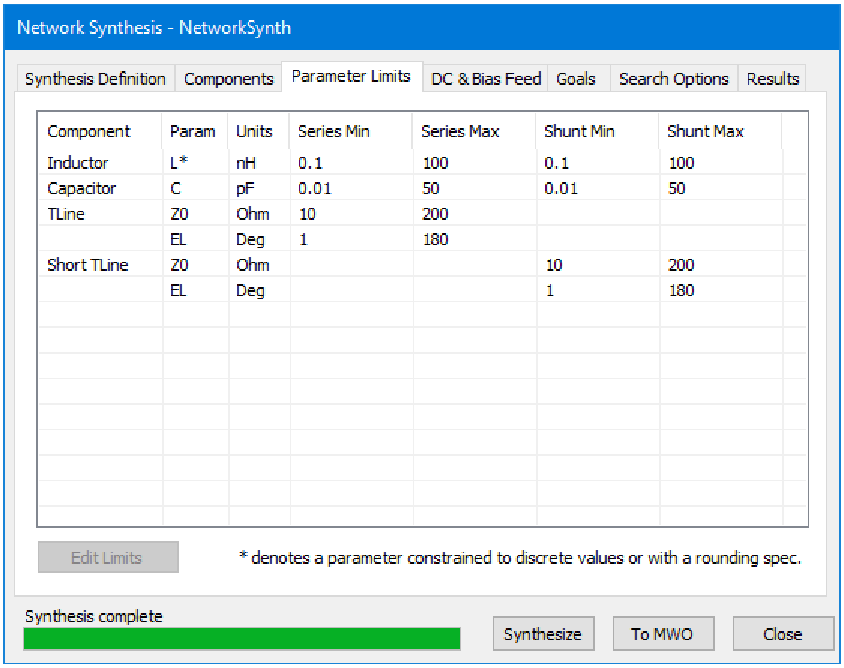
On the left of the dialog box a copy of the AWR Design Environment platform Elements Browser **Libraries** node displays. To select components, browse to a folder in the tree and then choose from the components in that folder listed in the pane on the right. **Shift**- and **Ctrl**-clicking are supported for multi-selection. Click on the **Add** button to copy the selected item(s) into the **Synthesis Components** pane on the right. This list of components is available to the wizard in the category (for example, **Capacitors**) specified in the drop-down list. The check boxes before each component name allow you to select which components can be considered for the synthesized networks.

The **Select All** and **Deselect All** buttons affect these check boxes. The **Remove** button deletes selected components from the list. When only a single component in the list is selected, the **Show Group** button is enabled. Click this button to view the category in the Elements Browser from which the component was added.

You can export the list of components on the right into a comma-separated values (CSV) file for later use by clicking the **Save to File** button. Click the **Load from File** button to reimport the list into another wizard session. You can easily edit these CSV files outside of the wizard with a spreadsheet program or text editor software.

13.10.3. Parameter Limits Tab

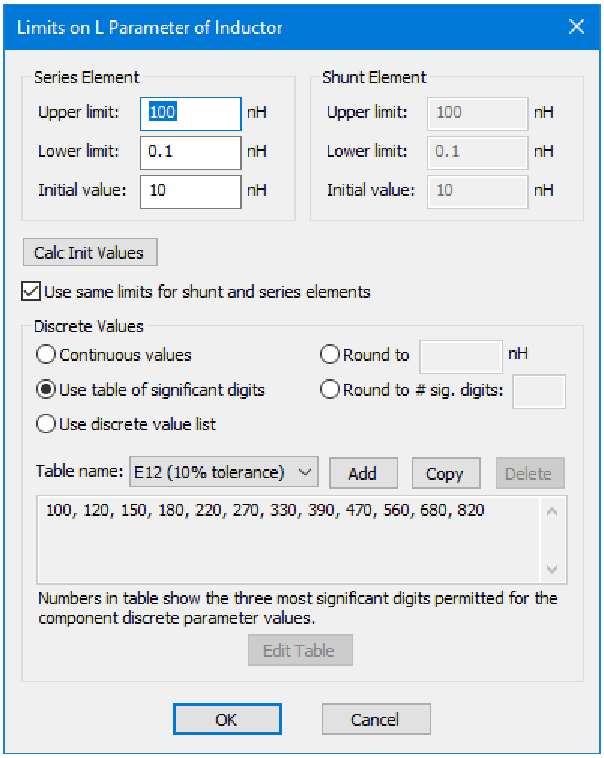
The **Parameter Limits** tab includes a table that lists each type of component selected on the **Components** tab.



Each component parameter shows the minimum and maximum values allowed when the component is used in either a series or a shunt configuration.

- If a specific component type is selected for the **First comp** or **Last comp** on the **Components** tab, a separate component type displays in this table, designated with "F/L" to indicate that it is a First or Last component in the network. This allows these First/Last components to have different limits than components of the same type used elsewhere in the networks.
- When **Replace TLines with MLINs** and **MSTEPs** are selected on the **Components** tab, a constraint on the ratio of adjacent MLIN widths, labeled as "W2/W1", also displays, with default min and max values set to the hard limits imposed by the MSTEP model.

To edit any of the constraints on a parameter, select it and click the **Edit Limits** button or simply double-click the parameter. A dialog box containing the various settings for the parameter constraints displays.



By default the **Upper**, **Lower**, and **Initial value** values for a parameter are the same for series and shunt components. To set the values independently, first clear the **Use same limits for shunt and series elements** check box.

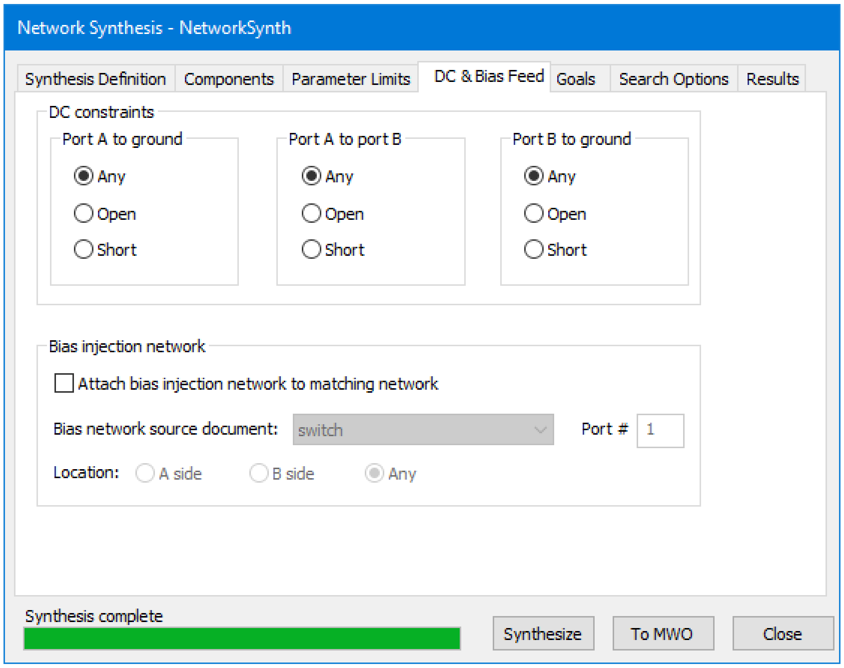
Click the **Calc Init Values** button to compute the initial value based on the upper and lower limits. Normally the geometric mean of the limits is used, but if the lower limit is set to zero, the arithmetic mean is used instead.

There are four options available for constraining the parameter to discrete values (if **Continuous values** is selected (the default), **Use discrete value list** is disabled):

1. Select **Round to**, then in the text box enter the precision in the displayed units to round to a specified precision. For example, to round the inductance parameter shown in the previous dialog box to the nearest tenth of a nH, enter **0.1** in the text box.
2. Select **Round to # sig. digits**, then in the text box enter the specified number of significant digits.
3. Select **Use table of significant digits** to enable the **Table name** drop-down and constrain the three most significant digits to those provided in a table. The names of built-in tables in the list are for the “E-series” system of preferred numbers, a standard (IEC 60063) that was created for use with electronic components. The three-digit values in the selected table display below the option and represent the values that are allowed for the three most significant digits of the parameter. User-defined tables are also supported. Click the **Add** button to create a table from scratch, and click the **Copy** button to create a new table using the values of an existing table as a starting point. User-defined tables are not defined only for a specific component parameter; they can be used for any parameter.
4. Select **Use discrete value list** and then select a **List name** from the drop-down list to constrain to a list of discrete values. Click the **Add** button to create a new list of allowed parameter values specified in base units, and click the **Copy** button to create a new list using the values of an existing list. The controls behave similar to when **Use table of significant digits** is selected, although there are no built-in lists of values.

13.10.4. DC & Bias Feed Tab

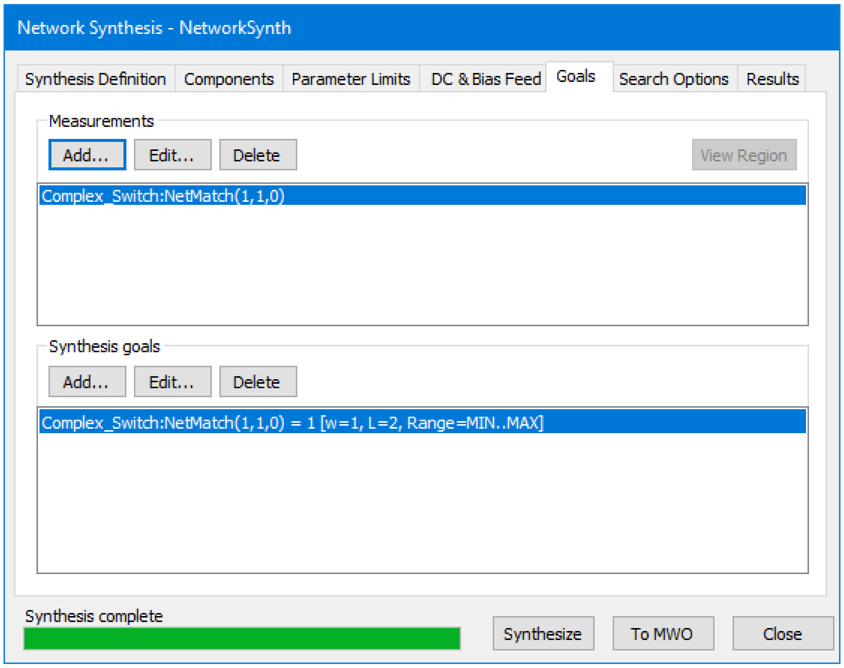
The **DC & Bias Feed** tab includes two sections. The top section has options for restricting the topologies to those that have certain DC characteristics (open or short from a port to ground or between ports). By default no DC constraints are specified. The bottom section provides options for attaching a bias feed network to the matching network.



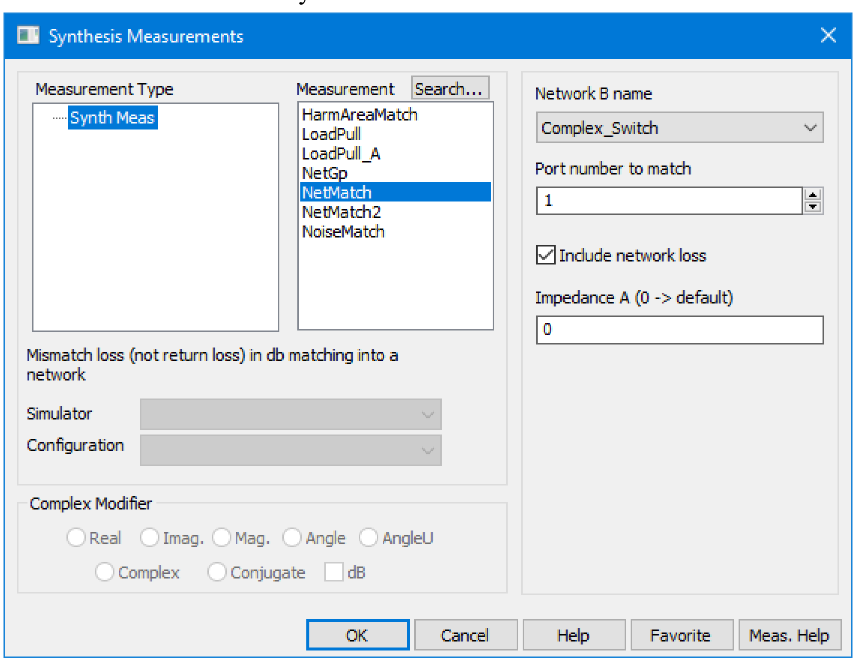
When **Attach bias injection network to matching network** is selected, the **Bias network source document** drop-down displays a list of the project documents (schematics and data files) that you can use for the feed network. The **Port #** option specifies which port of the feed network is connected to the matching network. For each matching network topology the wizard chooses a location to attach the feed network. If there is a DC constraint on the matching networks specifying that an open circuit is required between the two ports, you can also indicate whether the bias feed network should be located on the A side or the B side of the matching network.

13.10.5. Goals Tab

The fitness (or cost) of a matching network is evaluated by taking a measurement at each of the specified frequencies and summing comparisons of the measurements versus the goal. The definitions of the measurements are listed at the top of the **Goals** tab, and the goals for each measurement display in the bottom half of the dialog box.



In the **Measurements** section of the dialog box, click the **Add** or **Edit** buttons to display a Synthesis Measurements dialog box that lists the available synthesis measurements.

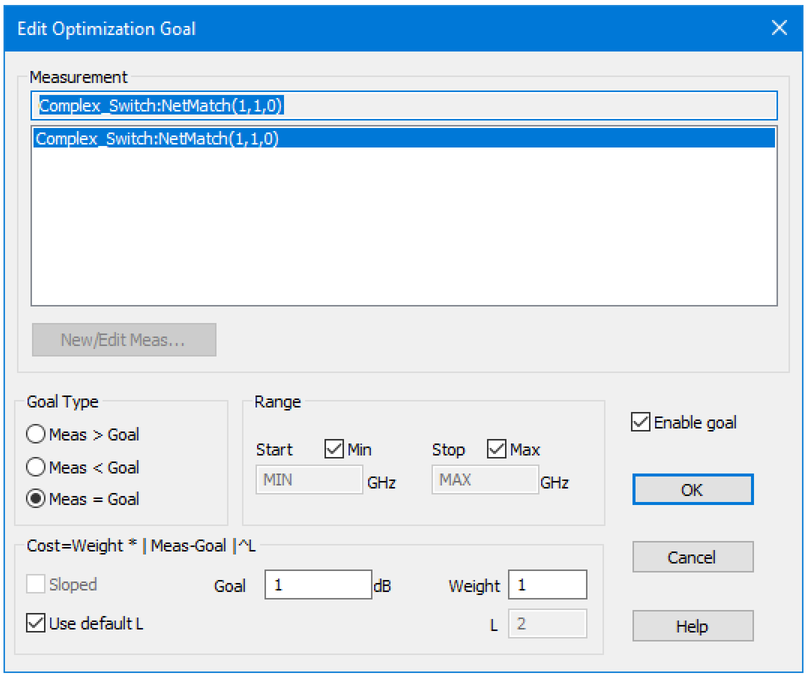


For synthesizing output matching networks for power amplifiers, the HarmAreaMatch and LoadPull measurements are useful. For low-noise amplifier input matching use the NoiseMatch measurement. The NetMatch measurement is good for input or output matching of a linear amplifier, and the NetMatch2 measurement provides a way to create an interstage match between two devices. (Note that the NetMatch and NetMatch2 measurements compute mismatch loss, not return loss.) You can use the NetGp measurement in conjunction with the other measurements to place a constraint on the amount of loss introduced in the matching network.

The HarmAreaMatch measurement provides a flexible way to directly specify a region (annular sector) of reflection coefficients to match into, at a specified harmonic. To aid in visualizing the region defined by the measurement parameters, when a HarmAreaMatch measurement is selected, the **View Region** button on the **Goals** tab is enabled. Click this button to display a Smith Chart with arcs drawn to show the boundaries of the region.

The CompCount measurement allows you to specify constraints on the number of components. When its **Filter Type** parameter is set to **Unique Vendor Lib Components** CompCount returns the number of different types of vendor library components used in the matching network. This can be helpful for minimizing the number of line items on a bill of materials. The other **Filter Type** setting is **Lumped Element Components**. When this mode is selected, the measurement returns the count of components in the network that are not transmission lines.

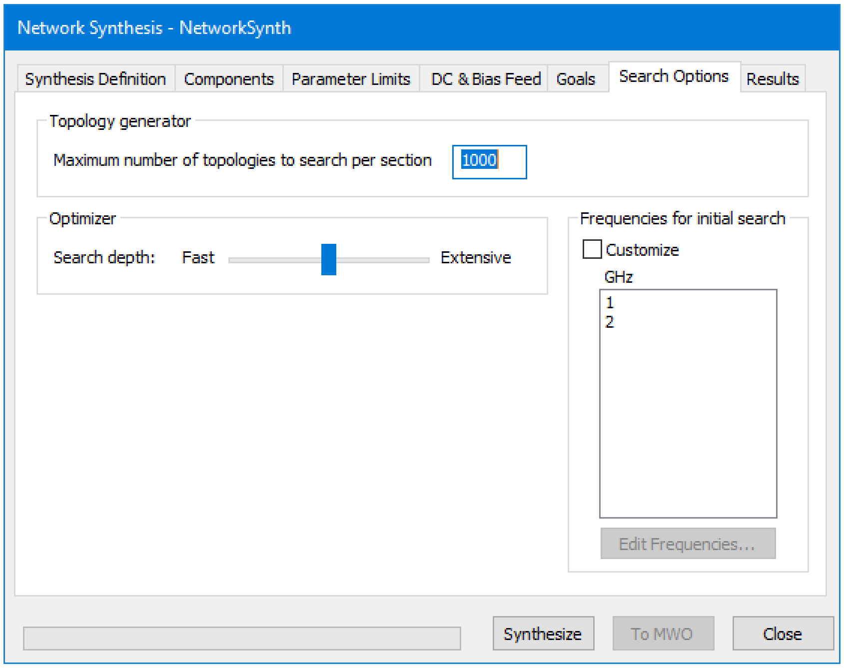
After defining the measurements, the goals are specified on the lower half of the dialog box. Click the **Add** button to display the New/Edit Optimization Goal dialog box to select a measurement and define a goal for it. The dialog box displays the formula used to compute the cost from the measurement and goal values. The values of the constants used in the formula are adjustable and you can alter the range of frequencies for the goal from the default MIN and MAX values, which correspond to the minimum and maximum frequencies listed on the **Synthesis Definition** tab.



Note that you can create multiple goals for each measurement, which means for example that the frequency range can be split into multiple bands with different goals for each.

13.10.6. Search Options Tab

The **Search Options** tab provides advanced settings for refining how network topologies are created and optimized. Descriptions of each option follow.



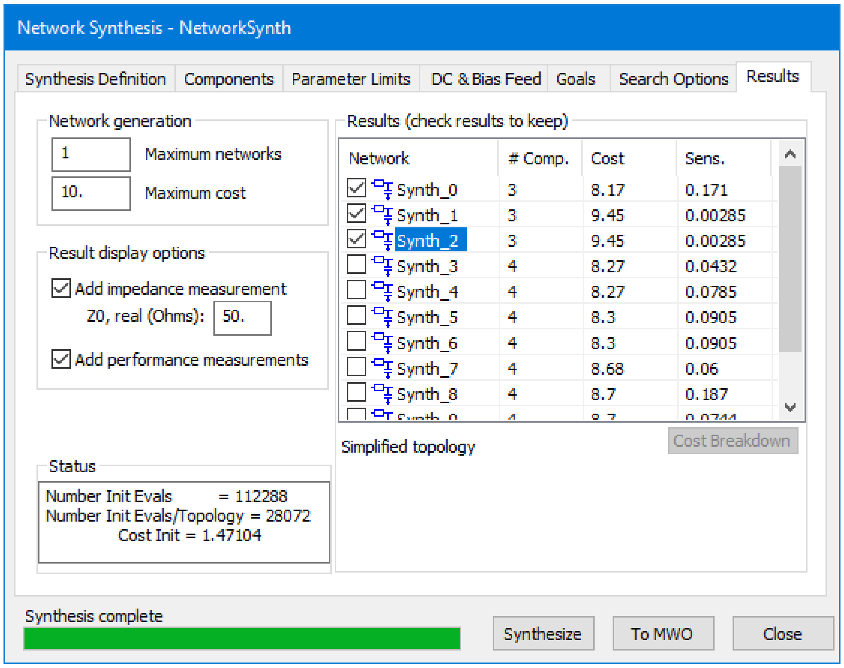
Maximum number of topologies to search per section: When the wizard generates the network topologies, it begins by creating all the possible topologies with a single component. The number of such topologies is normally equal to the number of selected check boxes on the **Components** tab. To create the 2-section topologies, it takes each 1-section topology and goes through the list of components that are allowed to follow the first one. The process repeats, producing an exponential growth in the number of topologies as a function of the number of sections. This setting (referred to as "M" and with a default of 1000) provides a way to constrain the exponential growth, by limiting the number of N-section topologies used to create the topologies with N+1 sections. Only the "M" best topologies are propagated.

Search depth: There are a variety of hard-coded control values in the algorithm used for optimizing the component values. These constants were determined empirically to provide a good tradeoff between covering the whole space of allowed values and limiting the optimization for speed and memory usage. This setting gives you some control over these optimization parameters.

Frequencies for initial search: During the initial phase of the synthesis process, when the full set of topologies is being evaluated for pruning, it can be helpful to use only a subset of the frequencies to speed up the evaluations. The list of frequencies on this tab is the subset used during this initial phase. The wizard chooses a default subset which you can override by selecting the **Customize** check box and clicking the **Edit Frequencies** button to specify alternate frequencies.

13.10.7. Results Tab

The outcome of the synthesis process is summarized on the **Results** tab.



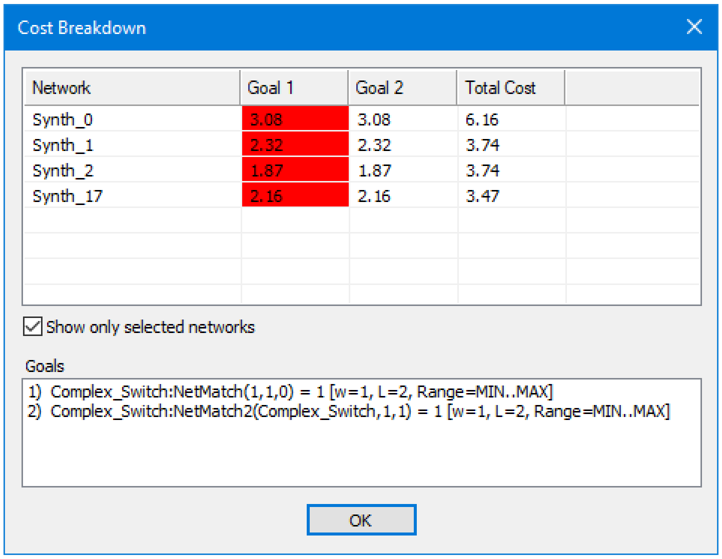
Listed under **Results** are all of the synthesized networks that have a cost less than or equal to the specified maximum cost, ordered by number of sections and cost. Click a column header to change the sort order. Networks with a lower cost are closer to meeting the goals. A network with zero cost achieves all of the goals– the synthesis procedure terminates if a network is found that attains a cost of zero. The **Sens.** column provides an indication of how sensitive the cost is to variations in the component parameters.

Click on a network name to see a schematic representation drawn in the **Simplified topology** area. If **Attach bias injection network to matching network** is selected on the **DC & Bias Feed** tab, a green arrow shows where the bias injection network is attached.

Click the **To MWO** button to export into project schematics those networks with a check mark before their name. By default, a certain number of networks (specified in **Maximum networks**) with the best costs are auto-selected.

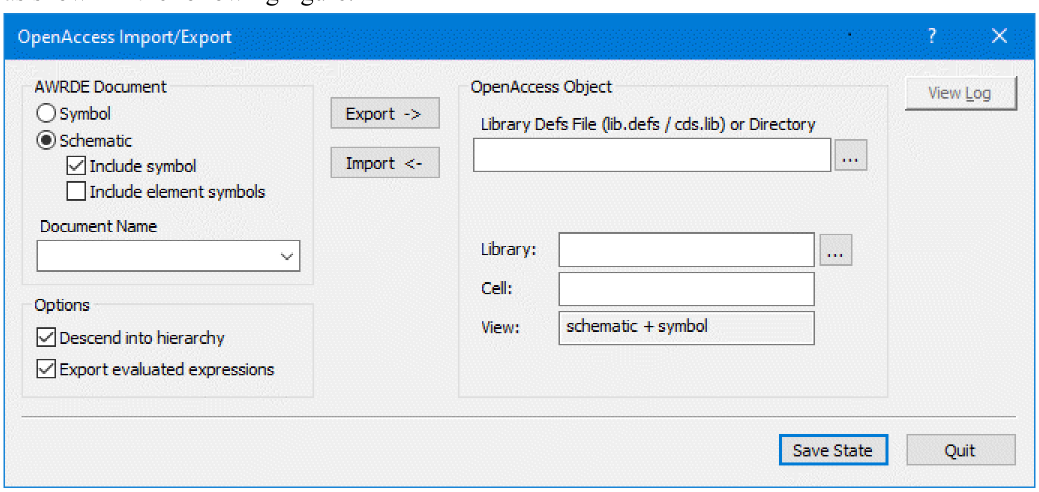
In addition, if one or both of the options under **Result display options** is selected, an Output Equations data display document is created in the project. This window will contain a schematic view and one or two graphs with measurements pre-populated. An S_TERM_Z measurement is used on the Reflection Coefficient graph, with the value for its Z0 (real) parameter specified in **Z0, real**. The network displayed in the schematic, which is also the data source for the measurements, is selected by clicking on the name of the schematic under the **User Folders** node in the Project Browser.

If there is more than one goal specified, the **Cost Breakdown** button is enabled. Click this button to display a Cost Breakdown dialog box that shows for each network how much each goal contributed to the total cost. For each network, the goal that contributed most to the total cost is highlighted in red and the goal that contributed least is highlighted in green.



13.11. OpenAccess Import/Export Wizard

The OpenAccess Import/Export Wizard allows you to import and export individual schematics and project or element symbols from/to Virtuoso and ADS OpenAccess databases. To access the OpenAccess Wizard, open the **Wizards** node in the Project Browser and double-click **OpenAccess Import/Export**. The OpenAccess Import/Export dialog box displays as shown in the following figure.



13.11.1. Specifying Options

In **Library Defs File or Directory**, specify either the full path to the *lib.defs* file or just the directory where the file is located (with a terminating backslash). If only the directory is specified and it does not contain a *lib.defs* file but does have an older-style *cds.lib* file, the wizard attempts to use *cds.lib*. Note that some statements (UNASSIGN, UNDEFINE, SOFTDEFINE, SOFTINCLUDE, and comments that begin with two dashes) supported in *cds.lib* files are not supported in OpenAccess. During export if the specified library does not exist, the wizard provides the option to create it. A *lib.defs* file will also be created in the specified directory during library creation if *lib.defs* or *cds.lib* does not already exist.

When you select **Descend into hierarchy**, any schematics and their associated symbols used by the selected schematic are imported/exported. For export, the schematics and symbols must be defined in the same project as the selected schematic. For import, the schematics and symbols must be in the same library as the selected schematic.

Export evaluated expressions applies only to export. When selected, expressions for parameter values are evaluated before they are exported (unless the expression contains schematic parameters). Otherwise, the expression is translated for the selected tool dialect.

13.11.2. Component Mapping

You can map the names of elements/models and the model parameters to/from ADS and Virtuoso via mapping files. A template map file, *OA_ADS_element_map.txt*, is provided in the *Wizards* directory, where *OpenAccessWiz.dll* is located. Place custom map files in the following locations:

- In a PDK the *.ini* file can point to mapping files, using lines such as the following:

```
[File Locations]
 OaAdsMap=Library\OA_ADS_element_map.txt
 OaVirtuosoMap=Library\OA_Virtuoso_element_map.txt
```

- In the *AppDataUser* directory there can be additional *OA_ADS_element_map.txt* and *OA_Virtuoso_element_map.txt* files for user-defined mappings.

The files are read in the order indicated above, and if there are multiple entries for a component in the different files, the entry read last takes precedence over any earlier entries. This allows you to override other maps by placing a map in *AppDataUser*.

The individual map entries in these text files provide a mechanism for specifying the library and cell name of the ADS/Virtuoso component and the name of the AWR Design Environment platform model or PDK element into which it should be mapped. Also, the names of parameters and their values can be transformed. The syntax for the entries is documented at the top of the example *OA_ADS_element_map.txt* file.

When exporting a schematic from the AWR Design Environment platform, if an element does not have a map entry it is written into the OA database with the cell name set to the original element name and the library name set to "awr_lib".

When importing a schematic into the AWR Design Environment platform, any components from an "awr_lib" library are created with models in the AWR Design Environment software having the same name as the OA cell name. Components not from "awr_lib" and with no map entry are converted into SUBCKT elements with the specified number of terminals. You can then specify the details in the template schematic to manually map the component. The SUBCKT element is also given a User Attribute, "oaLibrary", with the name of the OA library assigned to it. If the schematic is exported later, this lib name is used, and a Schematic View is not generated for the subcircuit.

Currently only these elements are automatically mapped:

AWR Design Environment	ADS	Virtuoso
GND	ads_rflib/GROUND	analogLib/gnd, basic/gnd
PORT	[shape: dot]	basic/ipin, basic/opin, basic/iopin
NCONN w/ Name="vss", "vdd", "vee", or "vcc"		analogLib/vss, vdd, vee, or vcc
K, INDM	ads_rflib/Mutual	analogLib/mind
NPORT_F		analogLib/nport

AWR Design Environment platform elements with dynamic symbols are an issue for export. In v13, NPORT_F is the only one that is specifically handled. The values of its "N" and "GND" parameters are included in the component name written out to ADS (for example: NPORT_F__N_3__GND_0). During import, anything in the component name after a double underscore is treated as an element parameter value specification.

13.11.3. Handling Variables

Variables defined in ADS Var components can be imported into the AWR Design Environment platform. If there are variables defined in an AWR Design Environment platform schematic, on export to ADS a Var block is created in the ADS schematic, but the variables must be added to the Var manually.

Virtuoso does not have a way to define variables in schematics (design variables are specified as part of the simulation setup), so the wizard does not transfer variables to or from Virtuoso.

13.11.4. Wizard Considerations

Note the following when working with the OpenAccess import/export process:

- Disabled elements are not exported.
- The values of enumerated parameters on components are stored as strings in the OA database, so the enumerated values of parameters in the two tools must have exactly the same set of strings.
- Unlike in the AWR Design Environment program, OA component parameters can be of Boolean type. In the AWR Design Environment program these parameters must be of enumerated type, with enumerations of "false" and "true".
- During export when **Descend into Hierarchy** is enabled, symbol views (without Schematic Views) are created for SUBCKT elements that reference data files in the AWR Design Environment platform project. When these are re-imported, the SUBCKT elements bind to the data files if they are already present in the AWR Design Environment platform project.
- Iterated (or vector) instances, buses, and bundles are supported. Taps can be used with buses and bundles (they are translated using NCONN elements in the AWR Design Environment software). The OA prefix and suffix repeat operators are also supported.
- PORT_NAME elements in the AWR Design Environment software are supported for export, provided that SUBCKTs that use them have schematic symbols with terminal names that match the PORT_NAME names. Connect-by-name parameters on SUBCKTs are not supported. PORT_NAME elements may be used with buses and bundles. On import, terminals that connect by name are converted to PORT_NAME elements.
- iCell notation (both normal and generalized) is supported for export. "iPar()" expressions in Virtuoso are supported for import.
- On export, INDK elements are converted to IND elements, and the schematics that define the IND elements in ADS and Virtuoso must contain an inductor with an ID of "L1". On import, inductors that are coupled are converted to INDKs.
- In the *lib.defs/cds.lib* file, the file path specified in a DEFINE may not contain space characters.
- If during an import with hierarchy, cells from different libraries have the same name, a conflict occurs in the AWR Design Environment program.
- Linux does not support spaces in library or cell names.
- the Microwave Office program does not support global nodes (vdd!). If they exist in a Virtuoso design you should convert the design to pass connectivity using named ports.
- Parameter frame locations are not preserved as part of the OpenAccess database.

- Linux paths and Cadence environment variables (for example, $CDSHOME) cannot be resolved by translator, so when directly reading cells from Linux you need an alternate *cds.lib* file with fully resolved UNC paths.
- Non-orthogonal and "virtual" wires in Virtuoso are converted to named connectors in the Microwave Office program.
- An NCONN or PORT_NAME connected to a GND element is not supported in Virtuoso.
- Variables are not sent to Virtuoso.

13.12. PCB Import Wizard

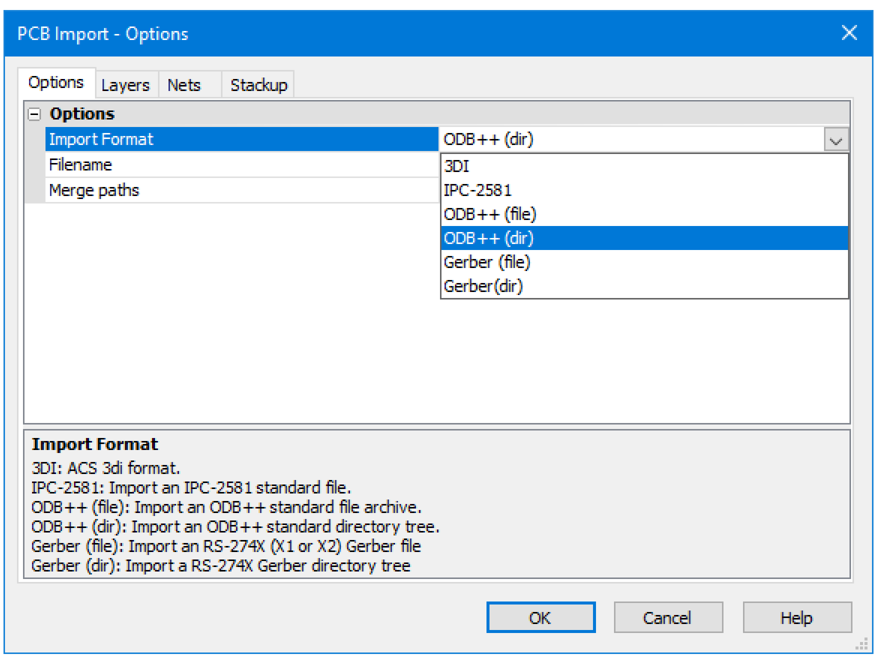
The PCB Import Wizard allows you to import 3Di, ODB++, IPC-2581, and Gerber (X1 and X2) standard files into the AWR Design Environment platform.

NOTE: Due to Allegro limitations, import of Allegro format files may fail when the AWR Design Environment platform installation directory contains *64bit* in the path name. You can install the AWR Design Environment platform in another location to avoid this issue.

13.12.1. IPC-2581, ODB++, and Gerber File Import

In addition to importing 3DI files, the PCB Import Wizard can also import IPC-2581, ODB++ (archived file or unarchived directory) and Gerber (archived file or unarchived directory) standard files. To use the PCB Import Wizard to import an ODB++, IPC-2581, or Gerber file, open the wizard and set the **Import Format** to the desired standard, then browse to the file using **Filename**. For more information about this dialog box, see “PCB Import Wizard Dialog Box: Options Tab”.

13.12.1.1. Supported Formats

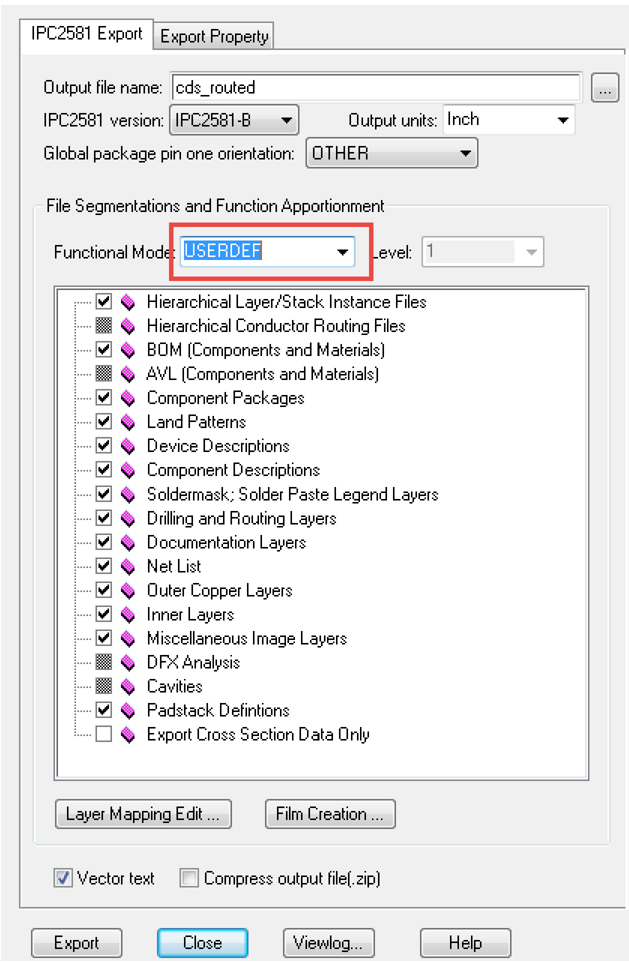


- IPC-2581 - supports files conforming to the IPC-2581 (A and B) standard. Common enterprise tools that support this format are Cadence Allegro and Zuken CR-8000.
- ODB++ (file) - supports files conforming to the ODB++ (V7 and V8) standard. These files are typically produced from Mentor Graphics tools.

- ODB++ (dir) - same format as ODB++ (file) except it operates on already uncompressed archives.
- Gerber (file) - operates on a compressed directory of Gerber files conforming to the X1 or X2 standard. A Gerber Job file is recommended to be part of the directory.
- Gerber (dir) - same format as Gerber (file) except it operates on an uncompressed directory of Gerber files. A Gerber Job file is recommended to be part of the directory.

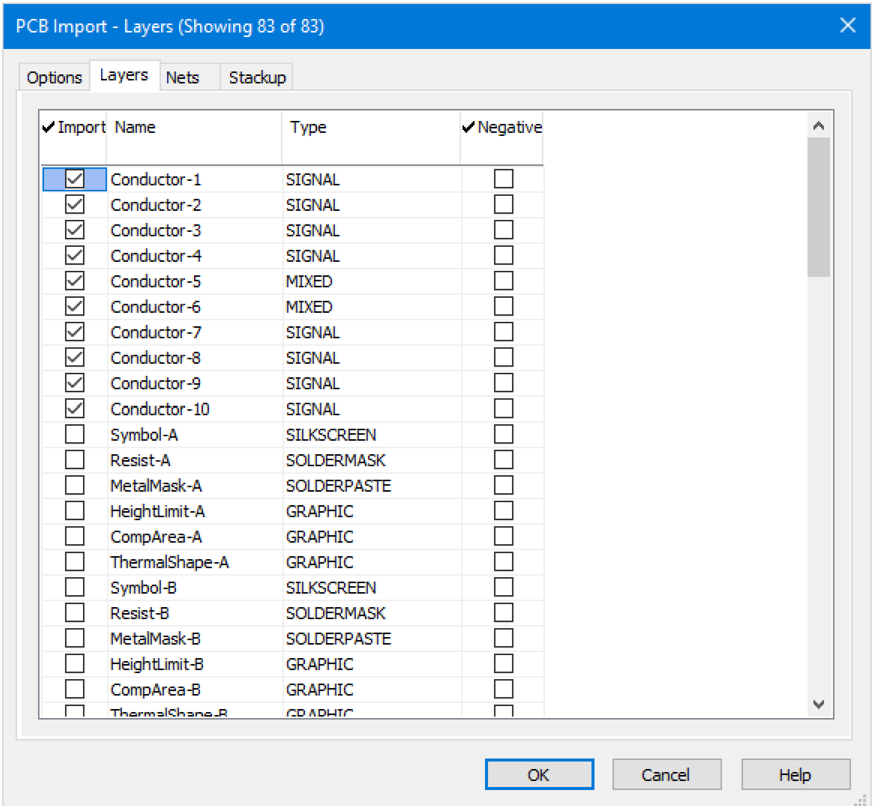
13.12.1.2. Exporting IPC-2581 from Allegro Software

When you export from the Allegro platform by choosing **File > Export > IPC2581** the **Functional Mode** must be set to USERDEF for import into the AWR Design Environment platform.



13.12.1.3. PCB Import Layers Options

After the correct format and file are selected on the **Options** tab, the import creates a new *.lpf* file containing the layer definitions that are specific to the imported PCB. If there is more than one STEP in the associated file, a unique *.lpf* is generated for each. You can view the layers on the PCB Import dialog box **Layers** tab.

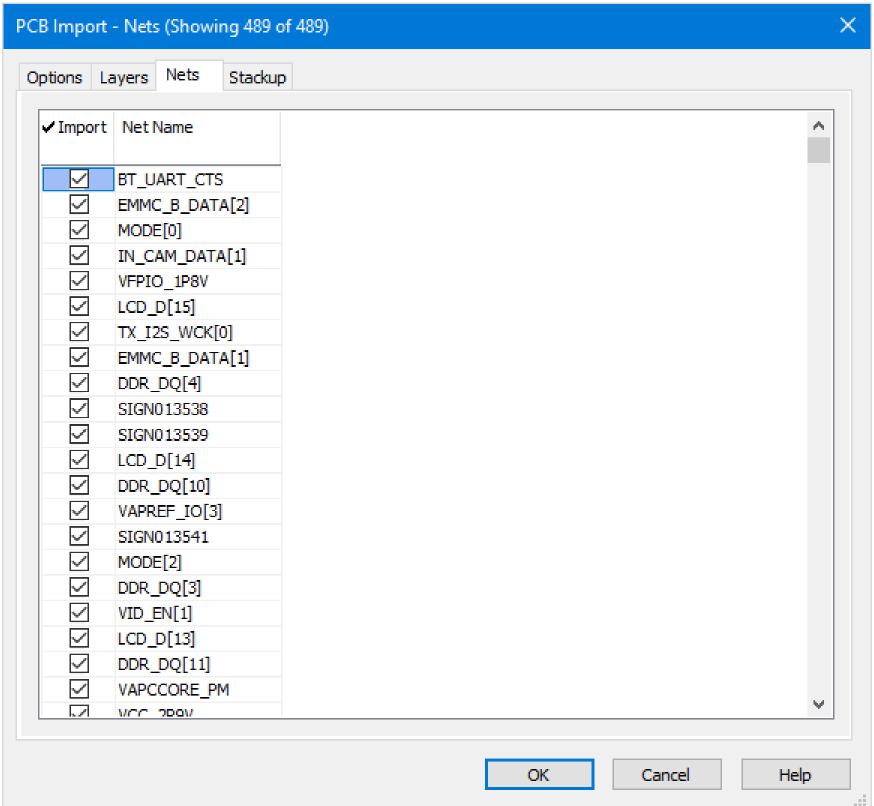


By default, only layers that contribute to the electrical portion of the design are imported. This selection is based on the Type of the layer. For IPC-2581, layers of type "CONDUCTOR", "PLANE", "SIGNAL", "MIXED", and "DRILL" are imported. For ODB++, layers of type "SIGNAL", "POWER_GROUND", "MIXED", "DIELECTRIC", and "DRILL" are imported. For Gerber "CONDUCTOR" and "DIELECTRIC" are imported. Selecting or clearing the **Import** check box next to the layer name determines whether or not it is imported. **Type** is for informational purposes, and **Negative** shows you which layers are negative layers. If you clear the check box, the layer is imported as positive.

All columns can be sorted by clicking the column title to toggle between no sorting, ascending, or descending. You can filter rows in the grid by typing a search string into the cell below the column title. The grid supports multi-selection using **Shift + Click** for range selection, **Ctrl + Click** for discontinuous selection, and **Ctrl + A** to select all cells in a column.

13.12.1.4. PCB Import Nets Options

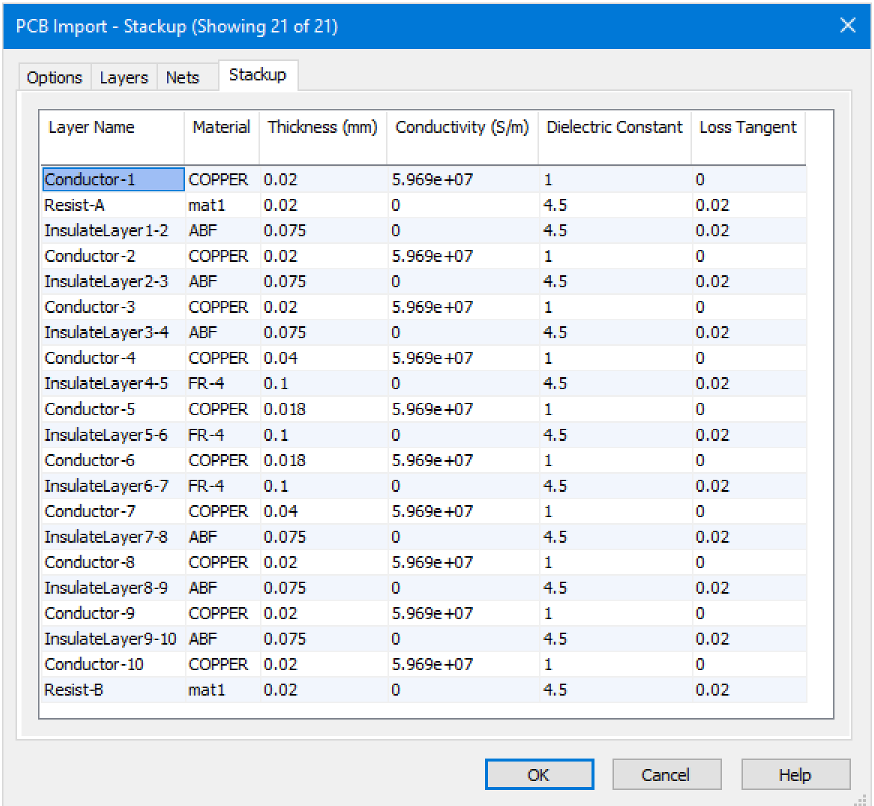
The **Nets** tab shows all of the electrical nets specific to the PCB design.



Individual electrical nets are included or excluded from import by selecting or clearing the associated **Import** column check box. Once selected, pressing the **Space bar** toggles the state. The grid supports multi-selection using **Shift + Click** for range selection, **Ctrl + Click** for discontinuous selection, and **Ctrl + A** to select all cells in a column.

13.12.1.5. PCB Import Stackup Options

The **Stackup** tab displays stackup information found in the design file or synthesized from the file.



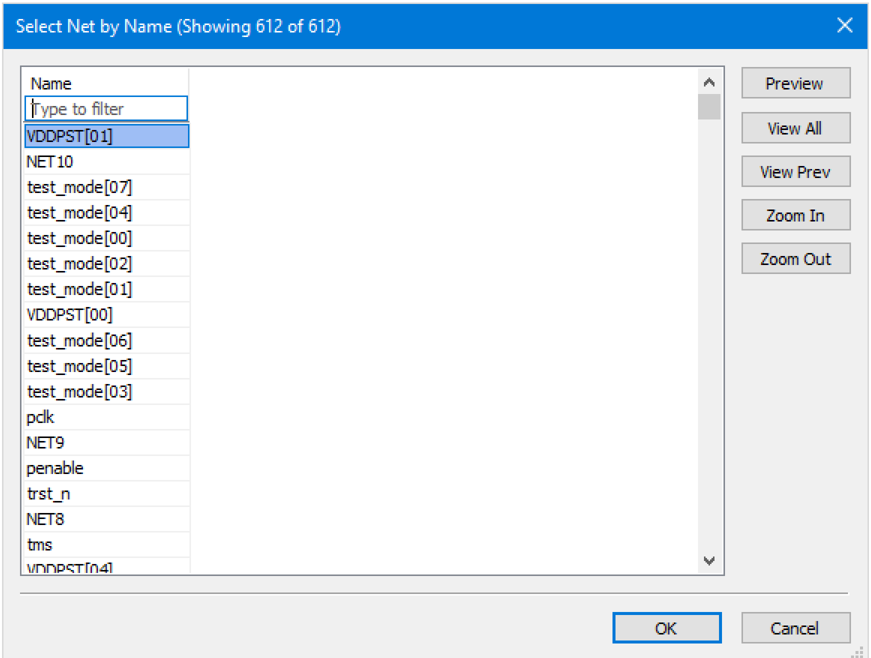
Not all files contain accurate stackup layer information like thickness, dielectric constant, or conductivity. If data is not found, N-1 dielectric layers, where N is the number of conductive layers imported, are added between the conductors and default values are used for missing data. All data can be edited, including multi-select editing support. This information is used to create a STACKUP element in the schematic. The STACKUP will have AIR dielectric layers added above and below the core. The thickness of these layers of AIR is equal to 1/4 of the total dielectric thickness of the core. Finally, the top and bottom boundaries are set to approximate opens.

13.12.1.6. PCB EM Setup Tool

The following sections describe the manual steps you can follow using the PCB EM Setup tool to import a PCB, select a region of the PCB, and copy that region to an EM structure. See PCB EM Setup help page for download and use instructions.

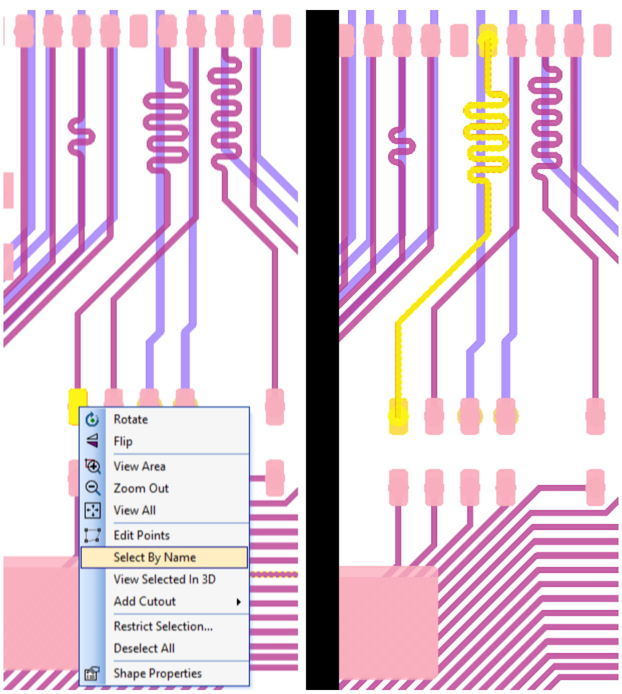
13.12.1.7. EM Structure Creation

After the PCB design is imported, you can select net names individually or as a group. All shapes with the same net name are considered to be part of the same electrical net, and the AWR Design Environment software can preserve these net names. Currently, net names do not drive connectivity but rather are present to aid selection by name. In a Layout View, choose **Edit > Select By Name** to display the following dialog box.



Choose one or more nets to select and click **OK**. Click **Preview** to zoom/pan to the selected net(s).

Alternatively, you can right-click a shape with a net name and choose **Select By Name** to select all other shapes with the same net name. This mode also supports multiple selected objects.

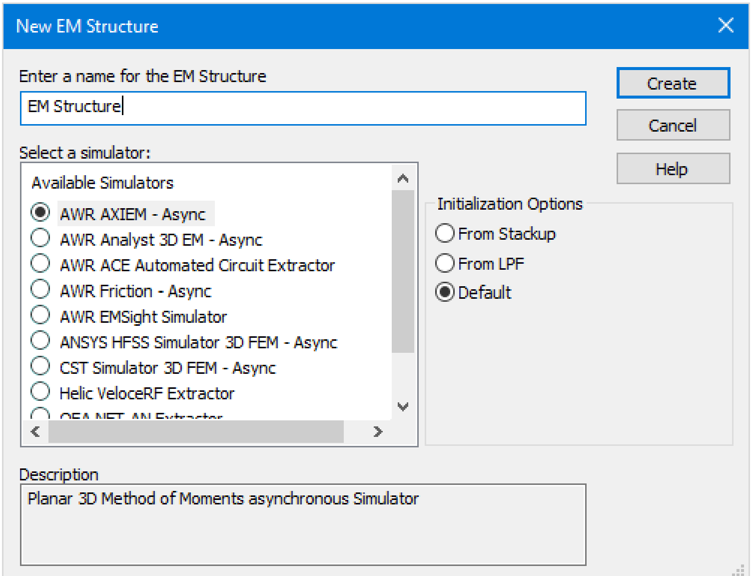


After you select the net(s) of interest, you can create an **EM Clip Region** and copy it to an EM structure.

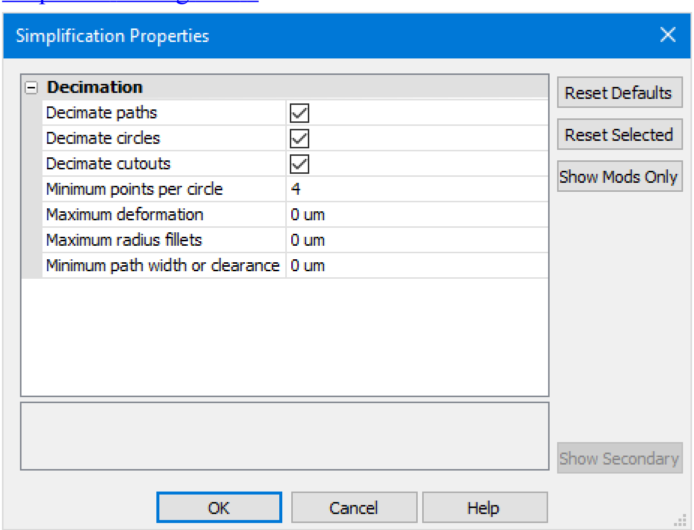
1. Select the shapes or clip regions, and choose **Layout > Copy to EM Structure**.

   The New EM Structure dialog box displays.

2. Select a simulator and desired **Initialization Options**, then click **Create**.



3. In the Simplification Properties dialog box, select the **Decimation Options** to apply, and click **OK**. See [“Simplification Properties Dialog Box ”]() for more information.



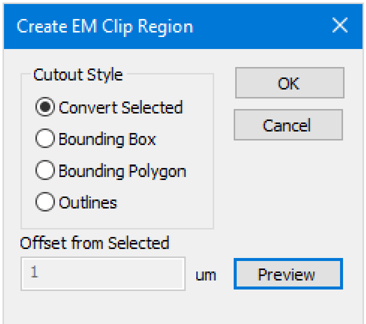
13.12.1.8. Trimming with EM Clip Region

EM Clip region allows you to trim a layout to manage its size and complexity for EM simulation. You can apply EM Clip region to only paths and polygon shapes in a schematic or EM layout.

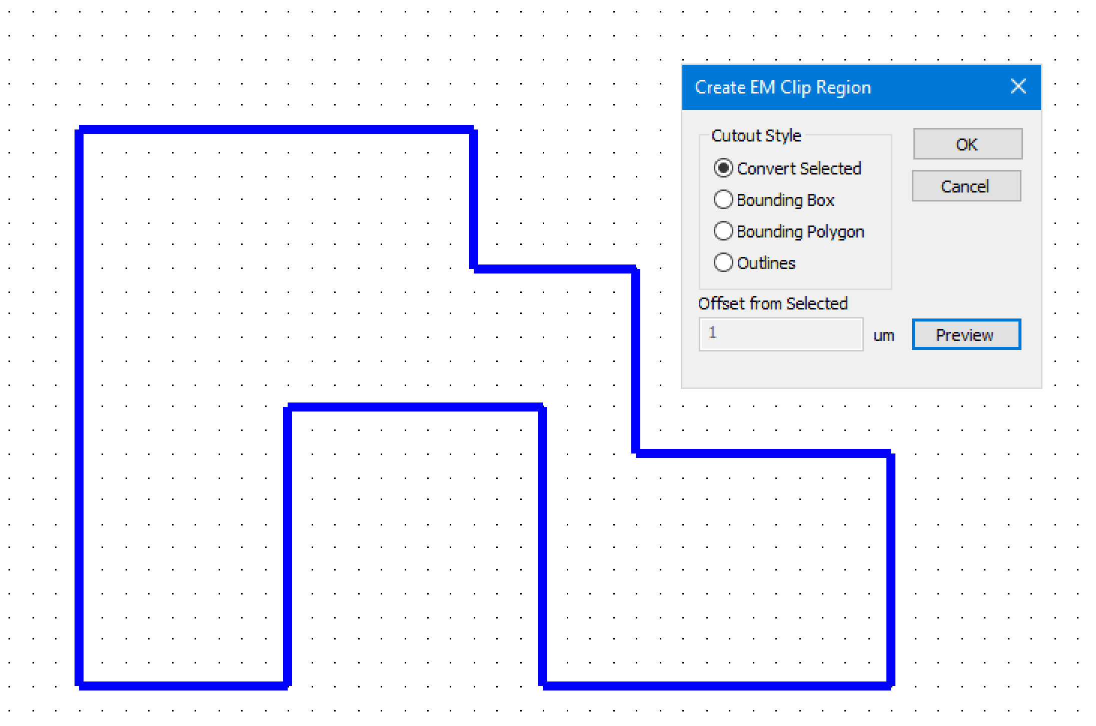
To draw an EM Clip region in a schematic or EM document:

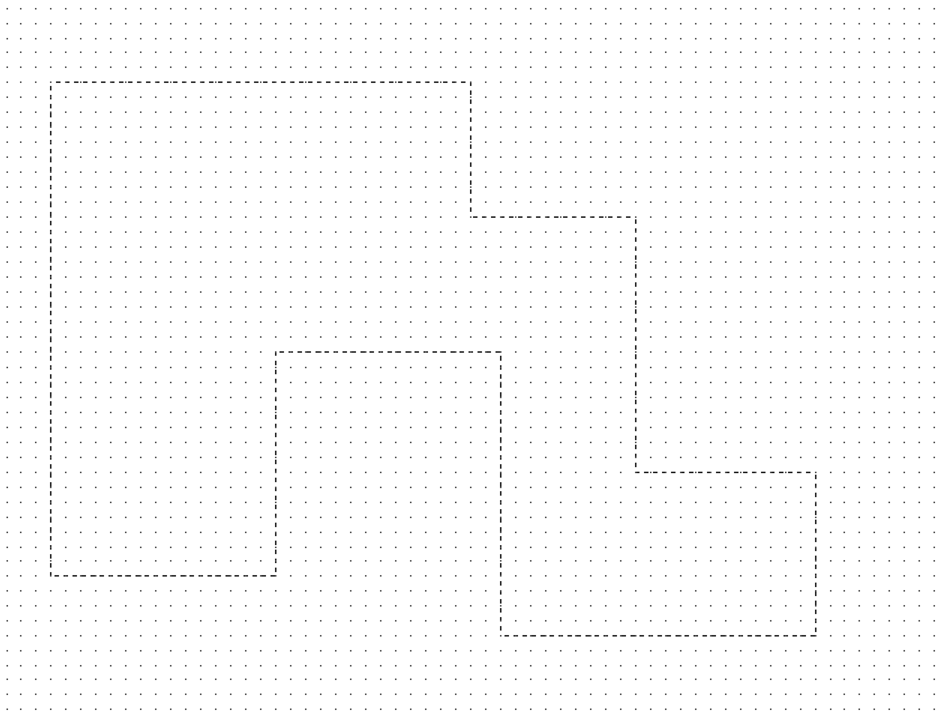
1. Select one or more shapes and choose **Draw > Create EM Clip Region**.

The Create EM Clip Region dialog box displays.

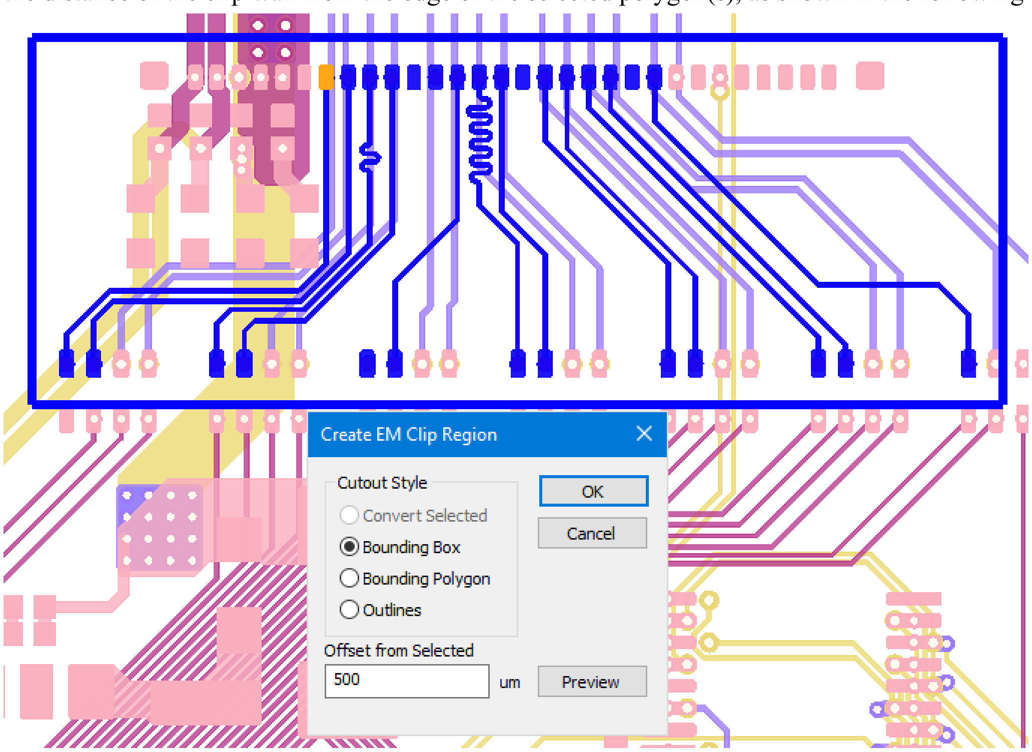


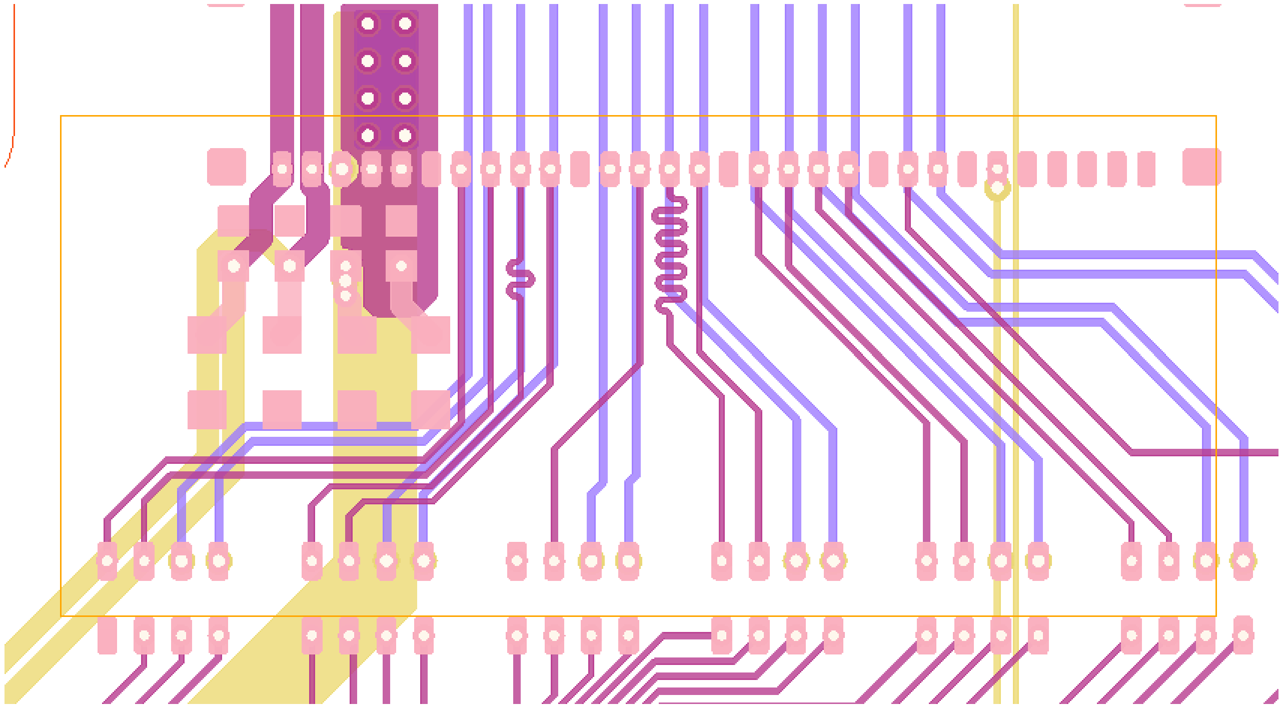
2. Select **Convert Selected** to convert a polygon shape to an arbitrary clip region, as shown in the following figures.



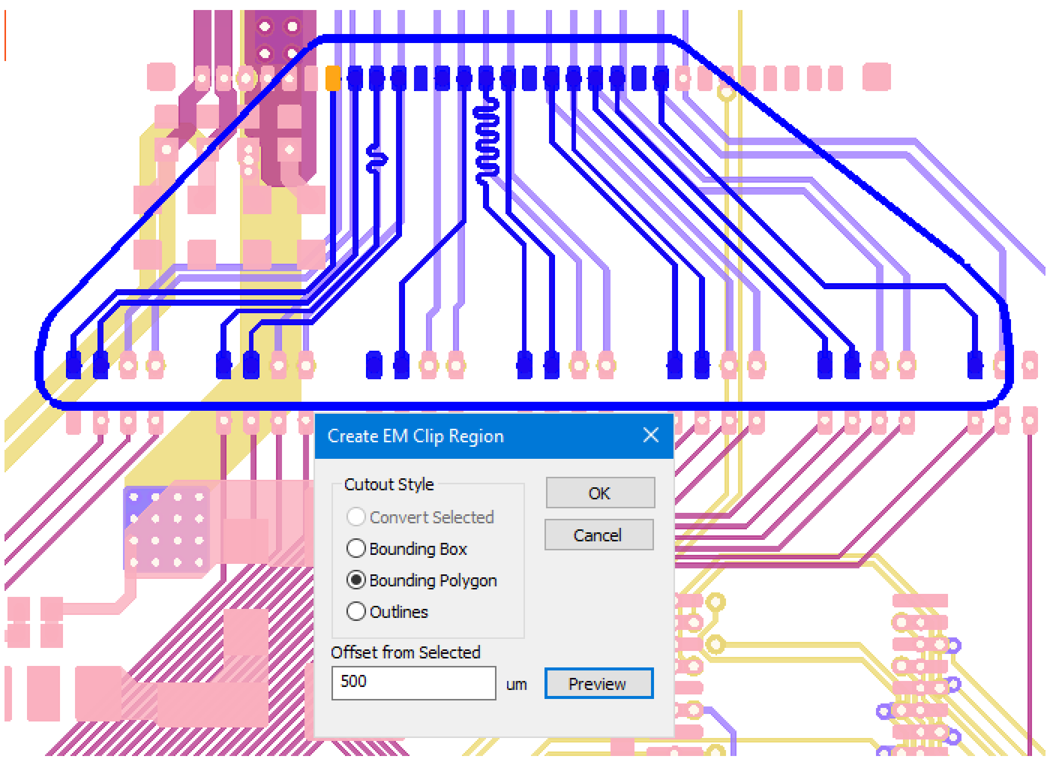


3. Select **Bounding Box** to draw a rectangular bounding box around the selected shapes. In **Offset from Selected**, specify the distance of the clip wall from the edge of the selected polygon(s), as shown in the following figures.

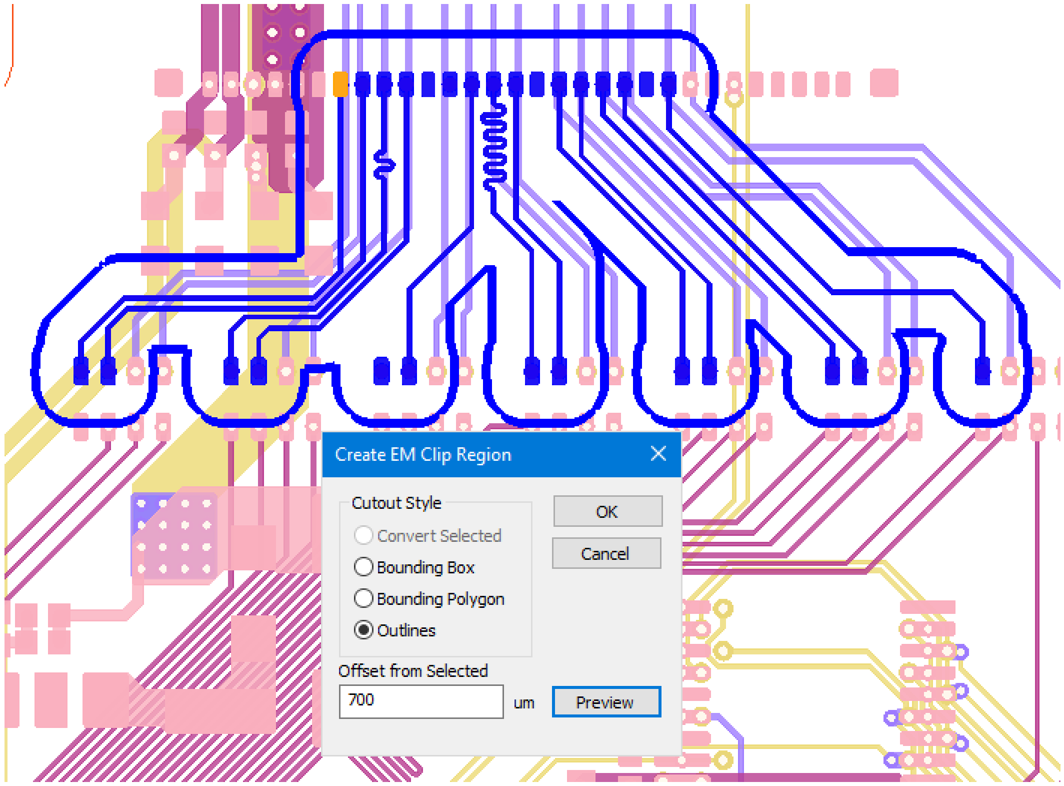




4. Select **Bounding Polygon** to create a clip region by following the outermost vertices of the polygon with a defined offset.



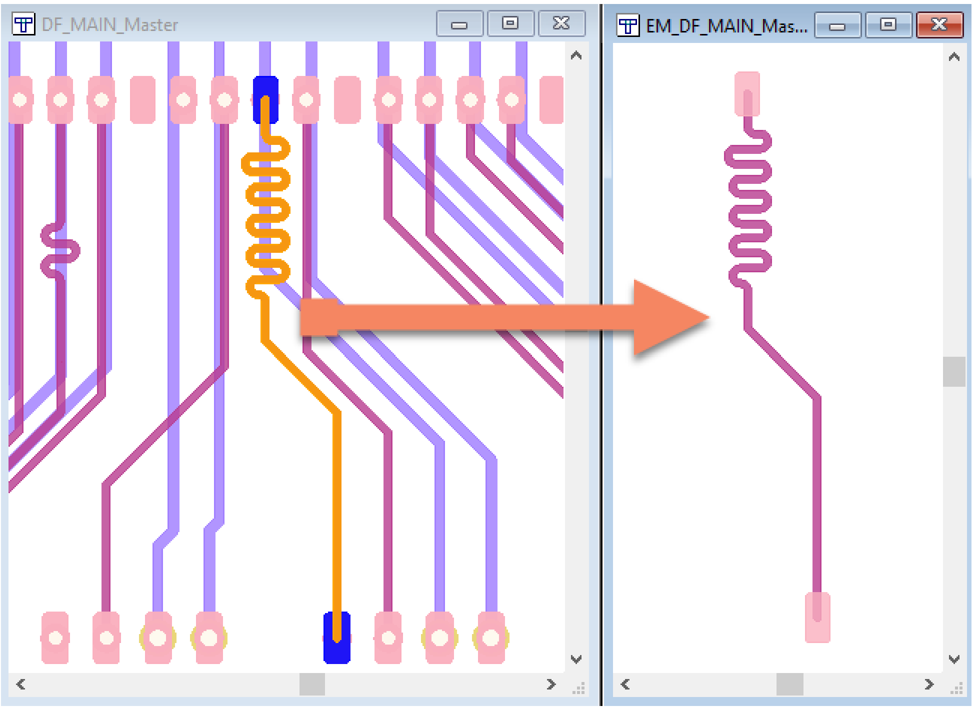
5. Select **Outlines** to joins the individual clip regions around the selected shapes if possible.



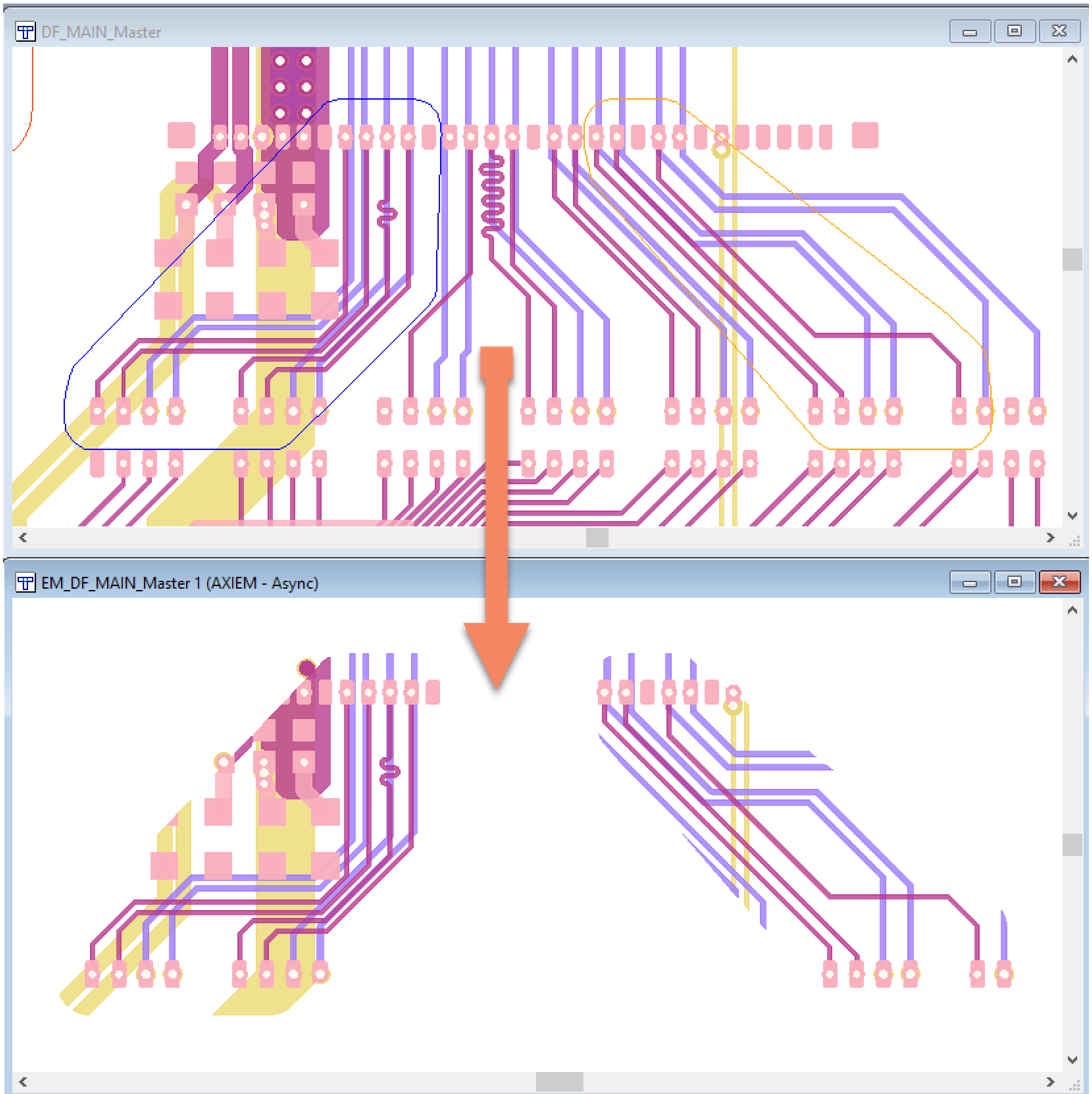
13.12.1.9. Clipping Shapes in Schematic Layout and Creating an EM Structure

To create an EM structure from a schematic layout, choose one of the following ways to send the shapes to the EM Structure:

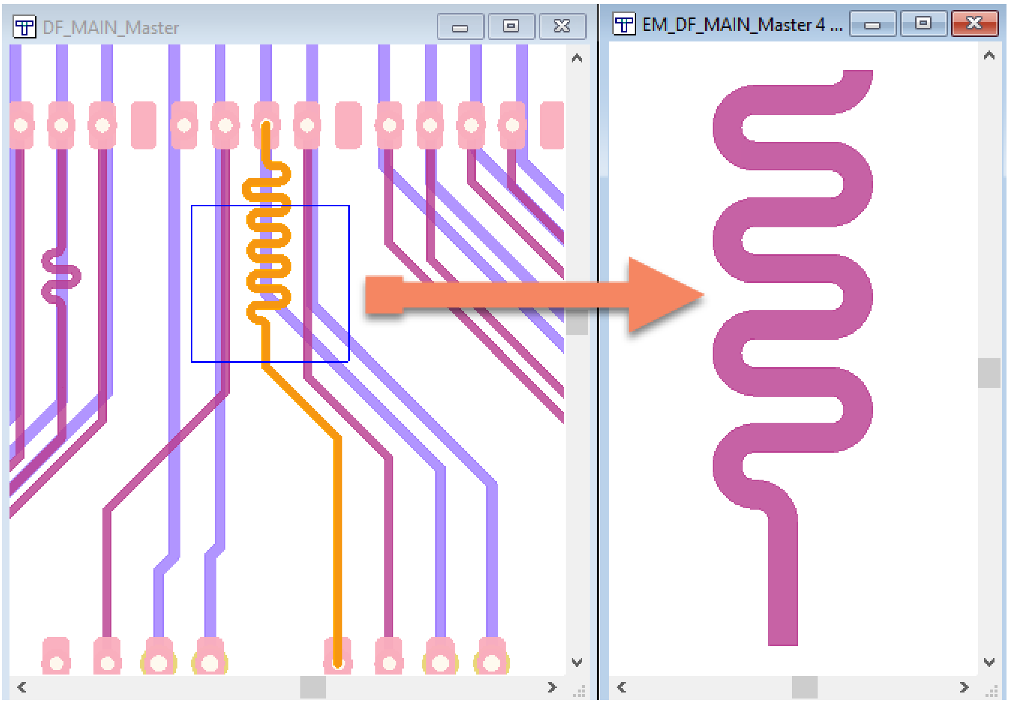
Select shapes only: Select only the shapes and copy them to the EM structure.



Select clip regions only: Select the clip regions only and the resulting shapes are copied to EM structure. You can select more than one clip region.



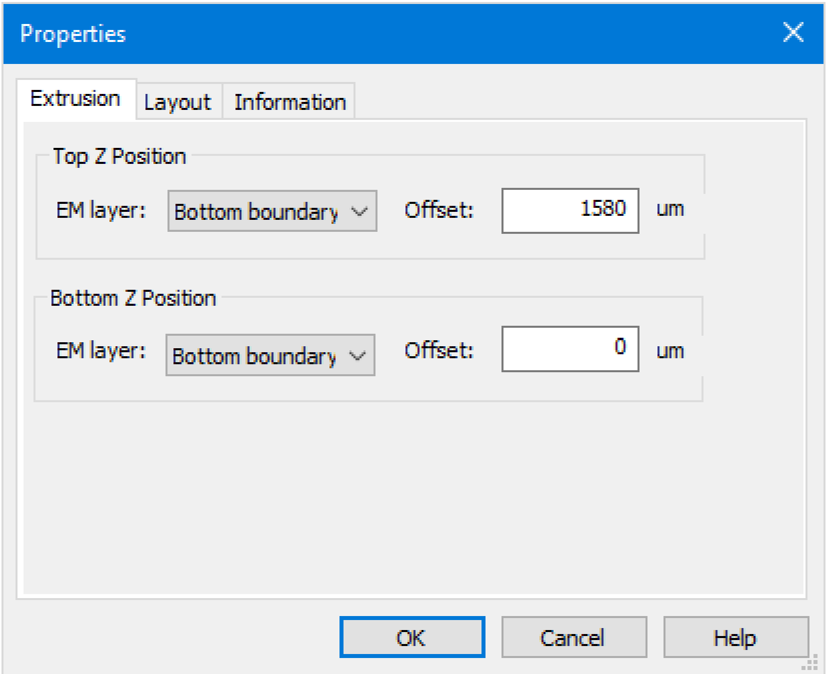
Select both clip regions and shapes: Select both the clip regions and shapes inside to copy the resulting shapes to the EM structure. Only the selected shapes inside the clip region are clipped.



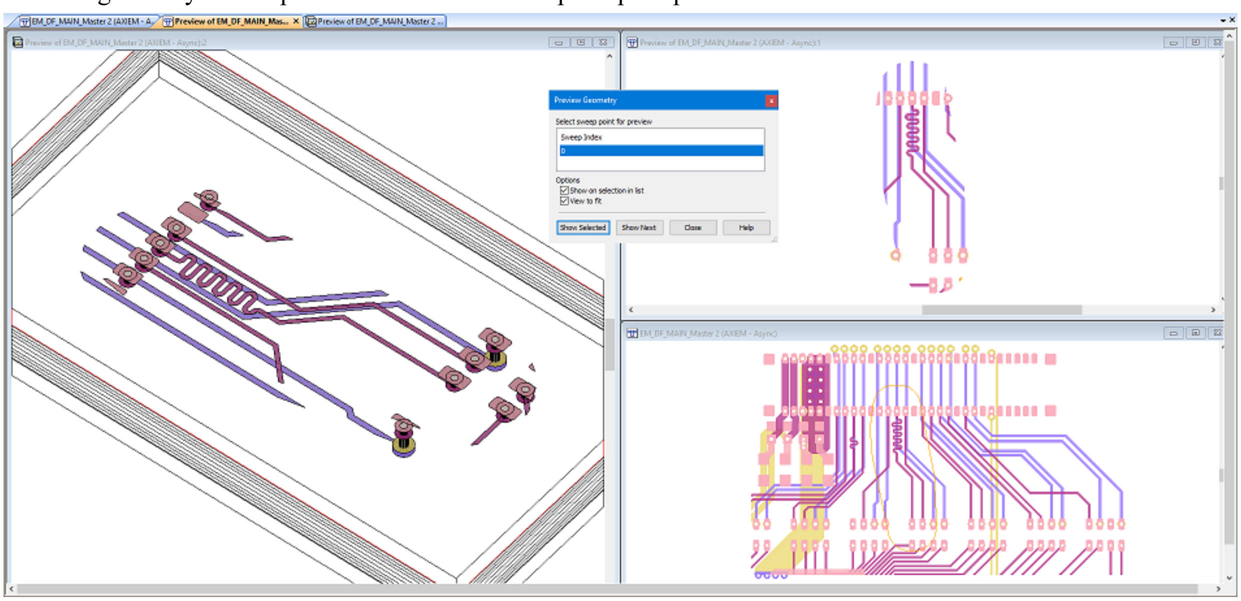
13.12.1.10. Editing EM Structure with Clip Region

You can trim EM structures by adding clip regions and then performing a simulation of the desired shapes only.

1. Clip regions in EM structure are drawn as described in "Trimming with EM Clip Region". Clip regions in EM structures operate in both X-Y and Z planes.
2. To clip the shape, choose **Draw > Modify Shapes > Clip Shapes**. You do not need to select a shape or clip area while performing this operation because it accounts for all the clip regions in the EM structure.
3. If necessary, set the Z dimension of the clip region by selecting the clip region, right-clicking and choosing **Shape Properties** to display the Properties dialog box. On the **Extrusion** tab, set the top and bottom Z position as desired. This is helpful when there is a multilayer EM structure but you only want to simulate a few layers.



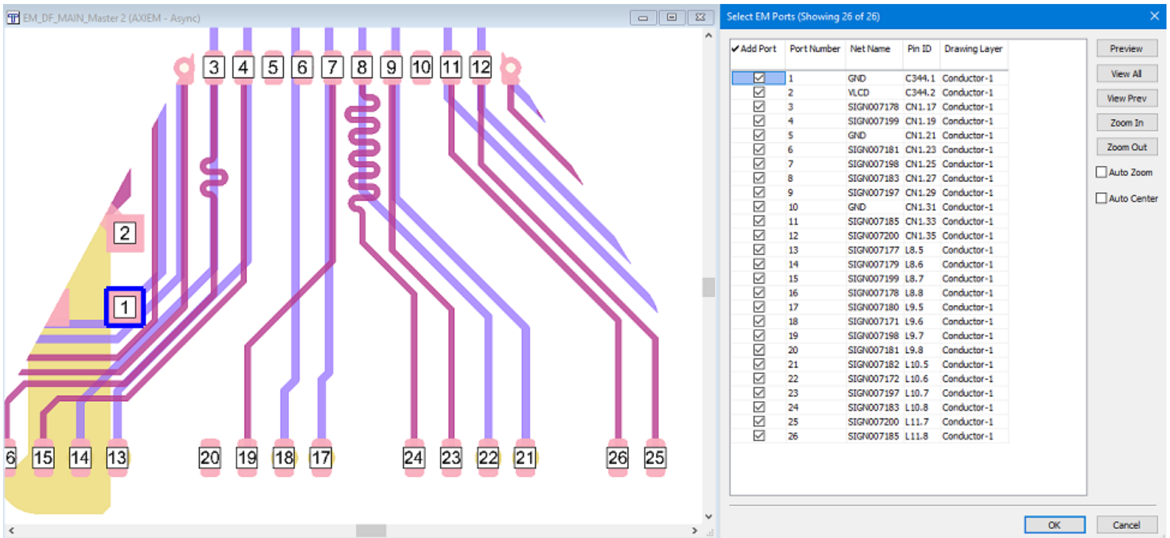
4. Preview the EM structure to ensure the desired clipping is performed along with the geometry simplification rules. Preview geometry can be performed without the Clip Shapes operation.



5. Perform an EM simulation if everything looks as desired.

13.12.1.11. Selecting PCB Pin Ports in an EM Structure

After the EM structure is created, if the copied geometry contained any pads identified as PCB pins, you can change them to EM ports by choosing **Draw > Create Ports from PCB Pins**. The Select EM Ports dialog box displays to allow you to add EM ports to existing PCB pins.

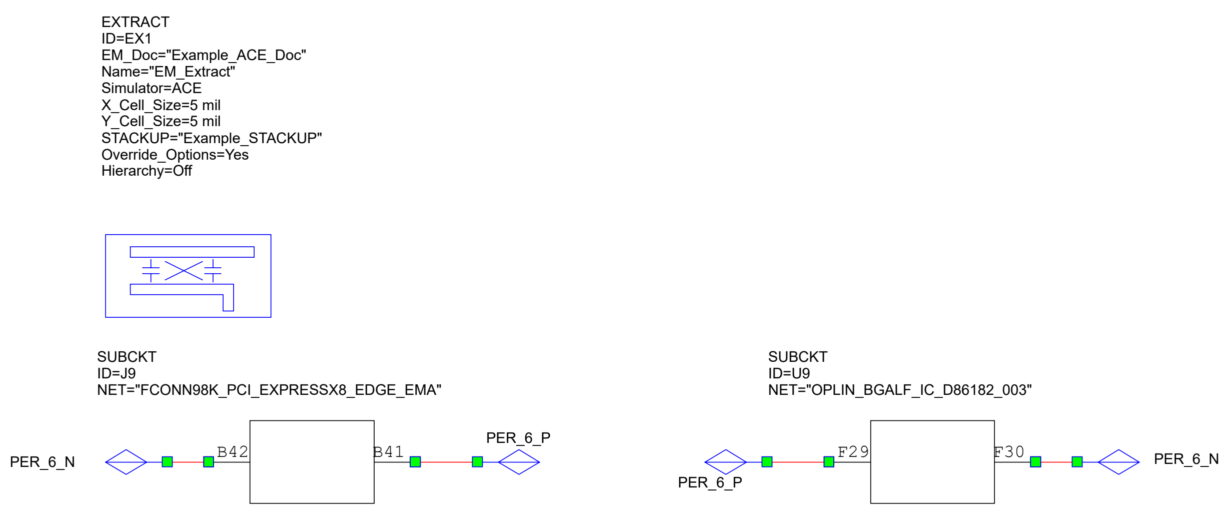


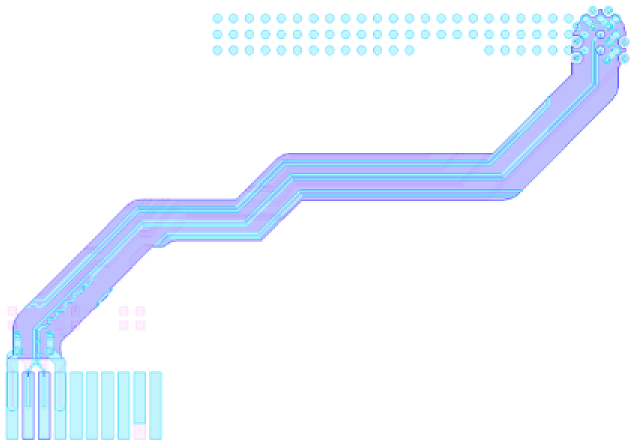
13.12.2. 3Di Import

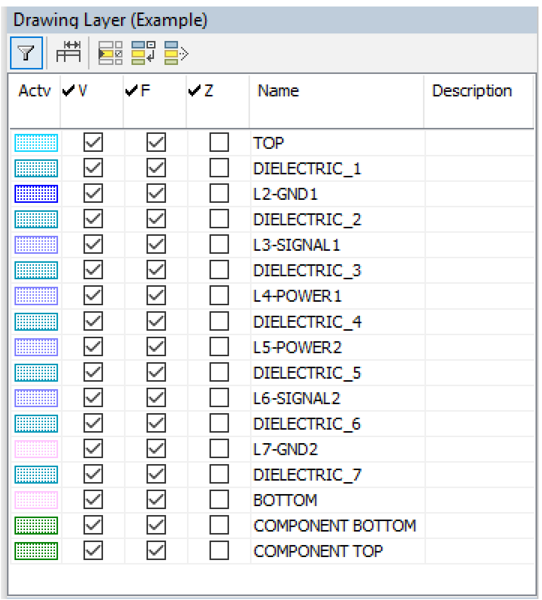
Imported 3Di files have a number of benefits:

- A schematic is created with placeholders for components.
- A layout is created using iNets to connect component footprints.
- Drawing layers and colors match the original database.
- A STACKUP element is created with all available dielectric information from the original database.
- The output document is ready to use the AWR extraction flow (see "EM: Automated Circuit Extraction (ACE)").

The following figures show examples of an imported schematic, layout, drawing layers, and EM STACKUP.

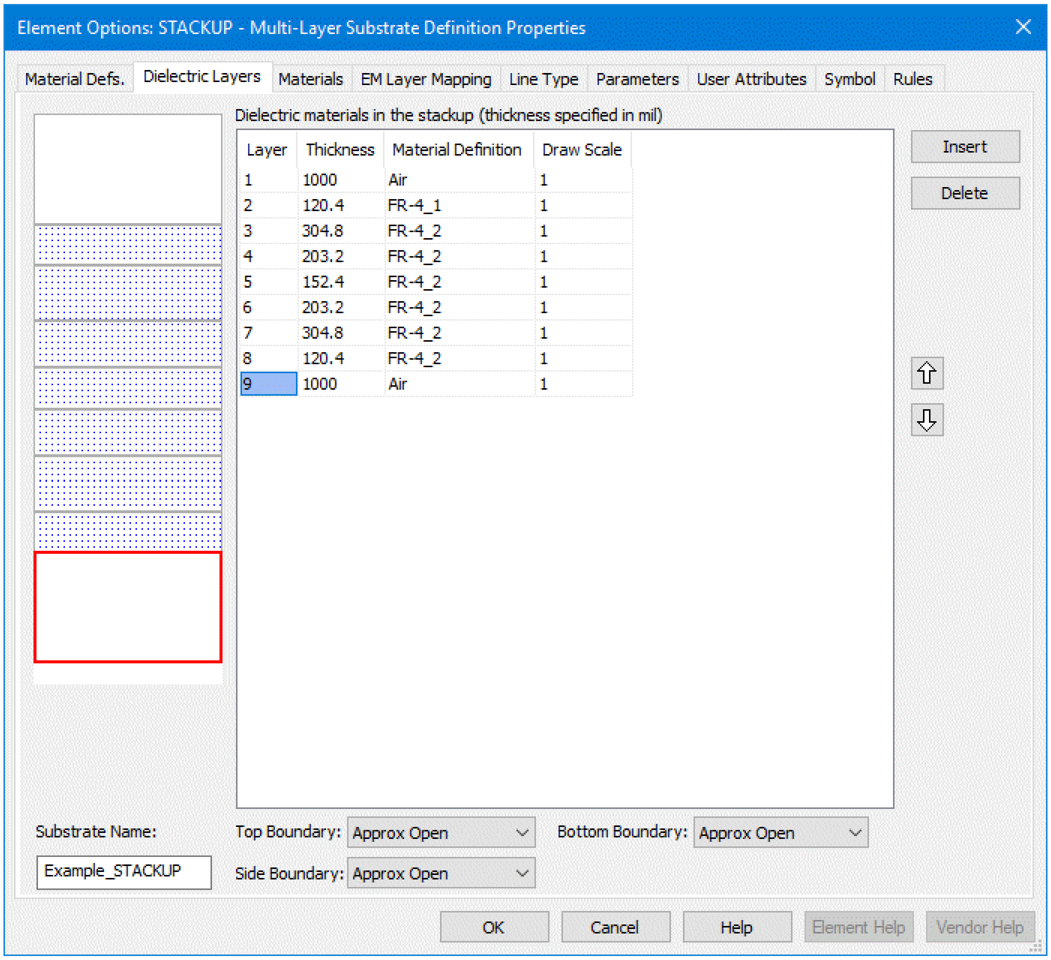






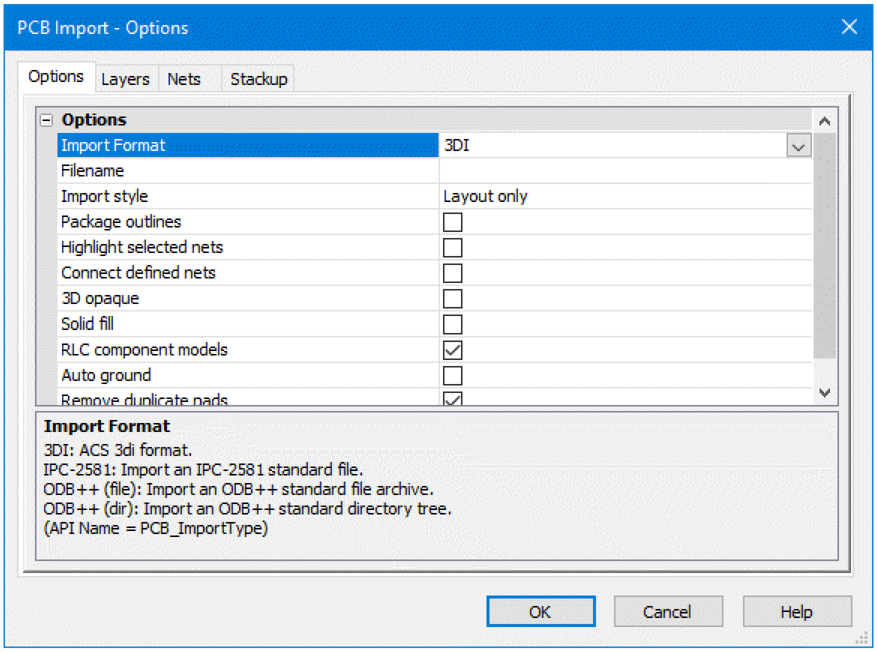
Drawing Layer (Example)

Actv	✔ V	✔ F	✔ Z	Name	Description
	☑	☑	☐	TOP	
	☑	☑	☐	DIELECTRIC_1	
	☑	☑	☐	L2-GND1	
	☑	☑	☐	DIELECTRIC_2	
	☑	☑	☐	L3-SIGNAL1	
	☑	☑	☐	DIELECTRIC_3	
	☑	☑	☐	L4-POWER1	
	☑	☑	☐	DIELECTRIC_4	
	☑	☑	☐	L5-POWER2	
	☑	☑	☐	DIELECTRIC_5	
	☑	☑	☐	L6-SIGNAL2	
	☑	☑	☐	DIELECTRIC_6	
	☑	☑	☐	L7-GND2	
	☑	☑	☐	DIELECTRIC_7	
	☑	☑	☐	BOTTOM	
	☑	☑	☐	COMPONENT BOTTOM	
	☑	☑	☐	COMPONENT TOP	



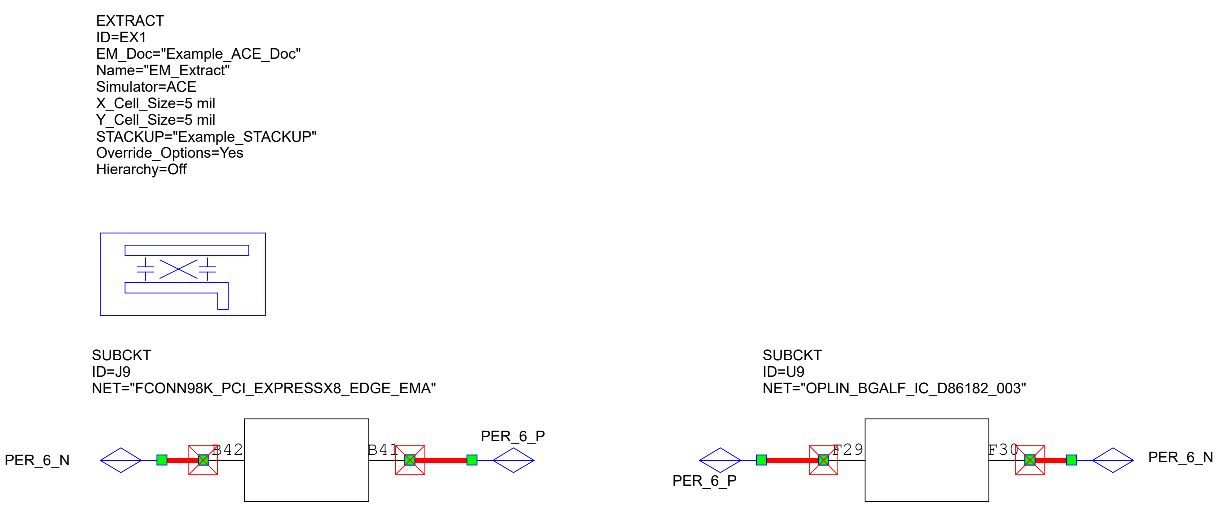
With the proper software license, you can run the PCB Import Wizard after downloading it from the Cadence website "Download Site" **Products** tab (www.awrcorp.com/download/login). After installation, to access the wizard, open the **Wizards** node in the Project Browser and double-click **PCB Import**. The PCB Import - Options dialog box displays.

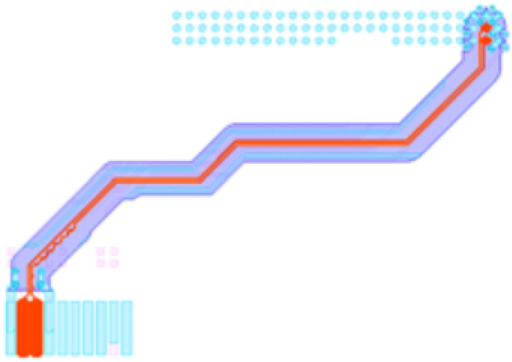
To import a *.3Di* file, set the **Import Format** to **3DI** and browse to the file using **Filename**. For more information about this dialog box, see “PCB Import Wizard Dialog Box: Options Tab”.



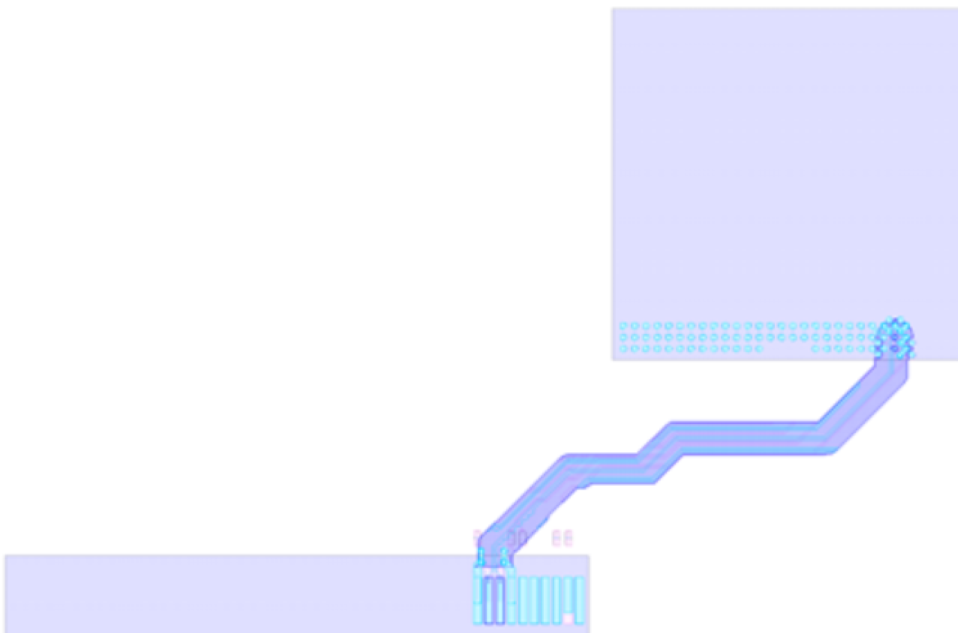
The following figures show commonly selected options.

Example circuit and layout with Highlight selected nets selected:



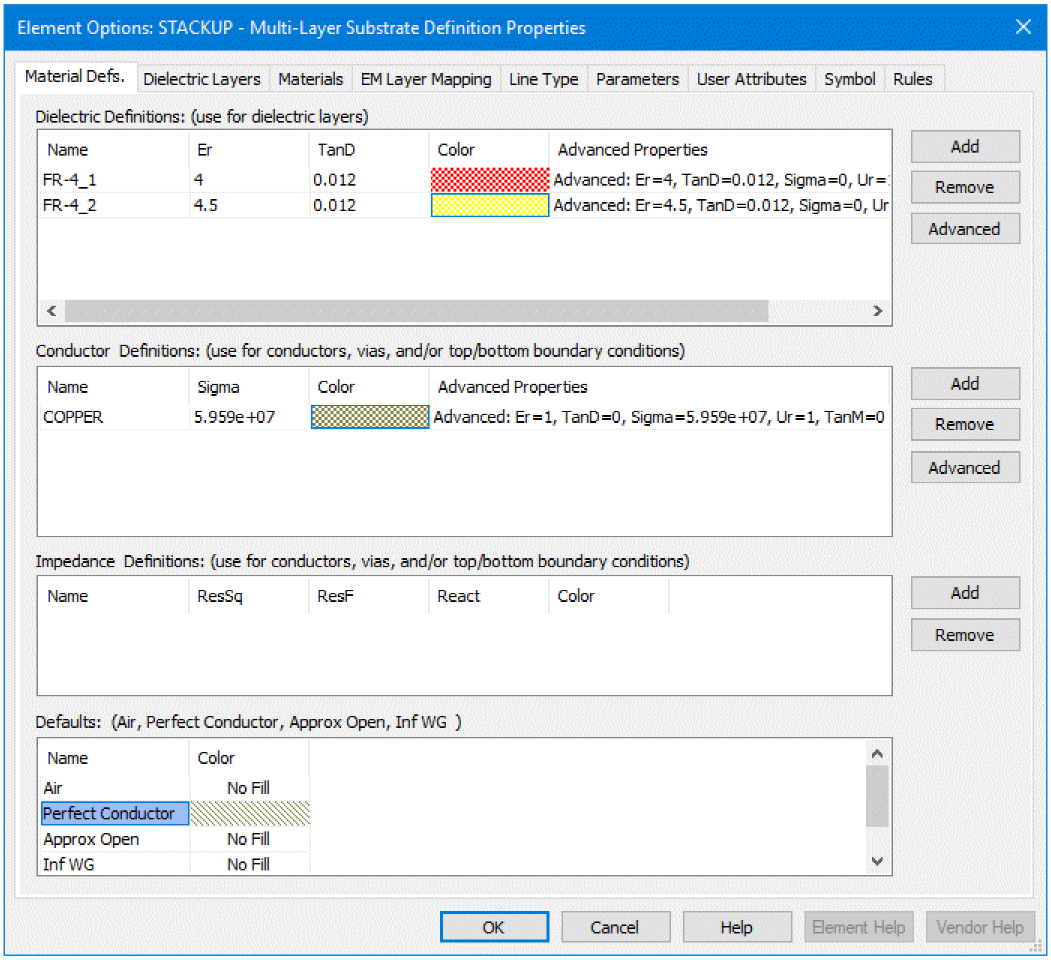


Example of whole component outlines, with Package Outlines selected:

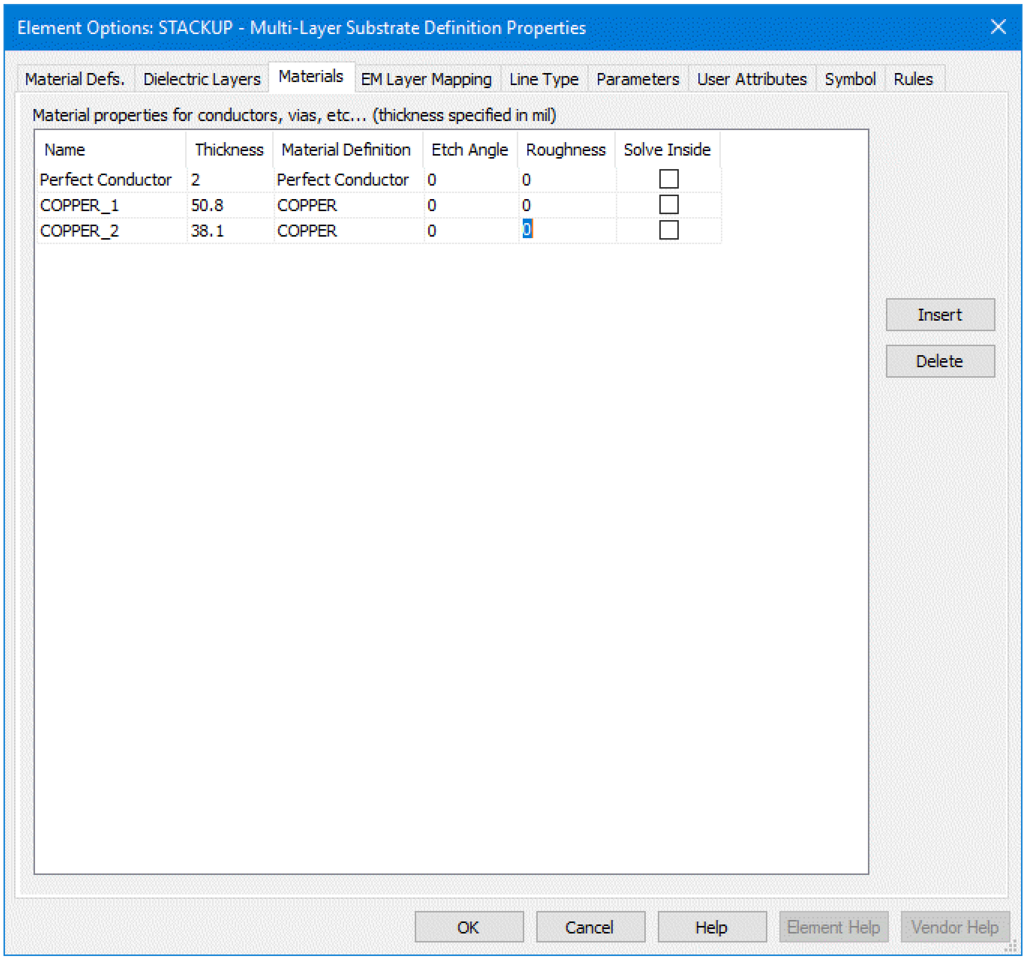


13.12.3. Dielectric and Conductor Information

The imported STACKUP has the dielectric and conductor information from the original database. You should verify these numbers for accuracy. The following figure shows the STACKUP material definitions.

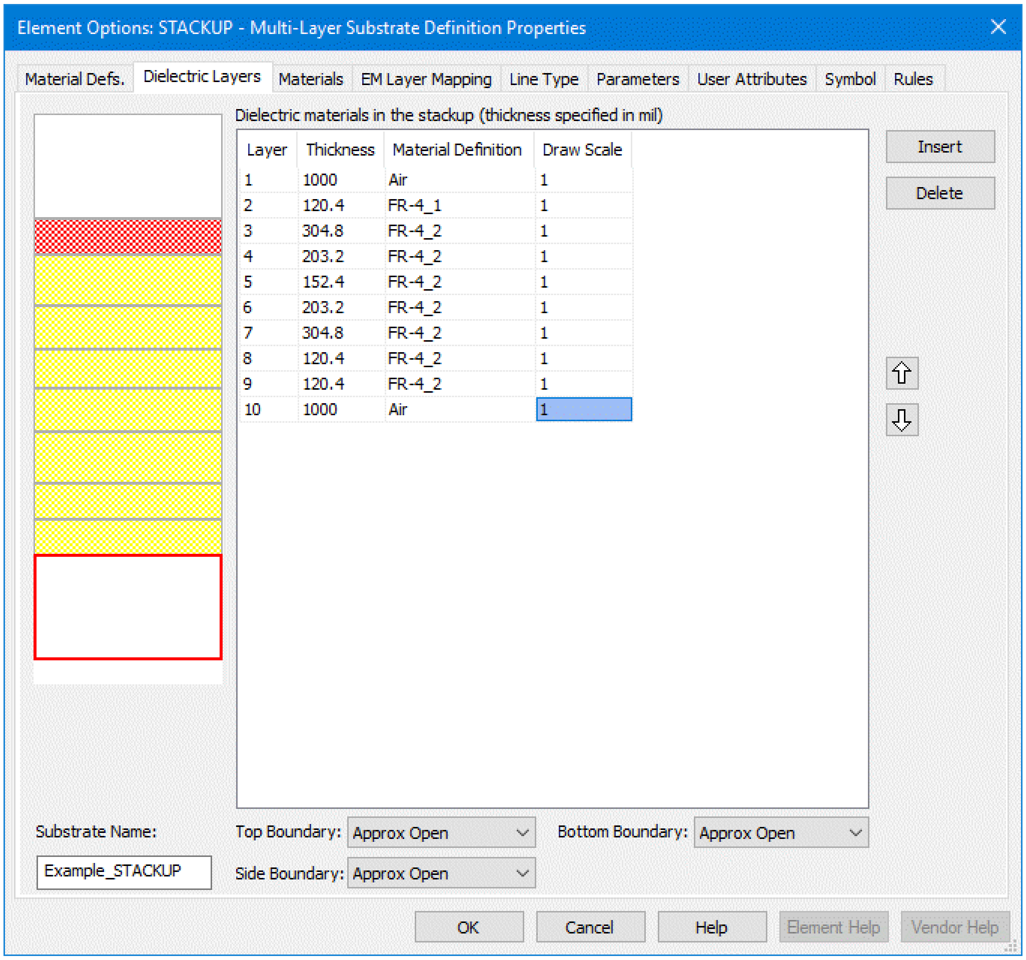


The following figure shows the STACKUP materials.



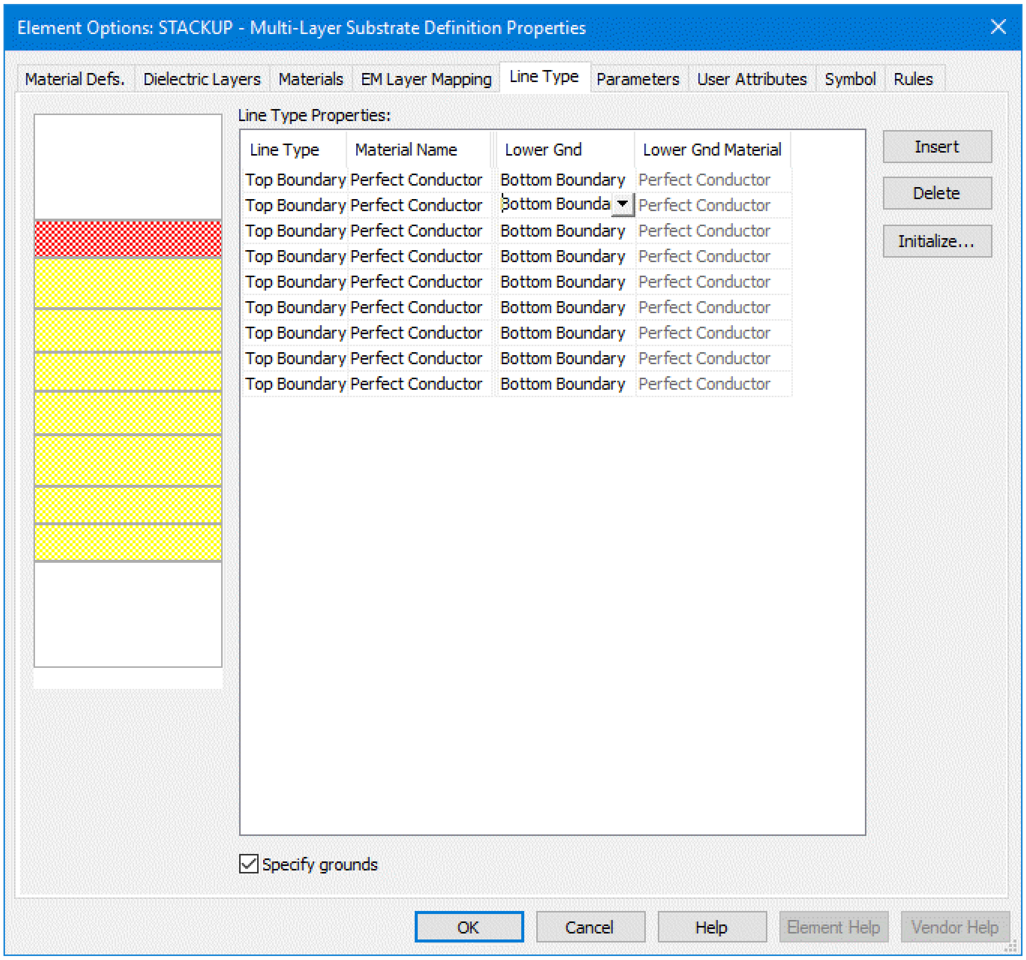
13.12.4. EM Boundaries

The imported STACKUP always has **Approx Open** set for both the top and bottom boundaries. You should verify this setting. The following figure shows the STACKUP boundaries.



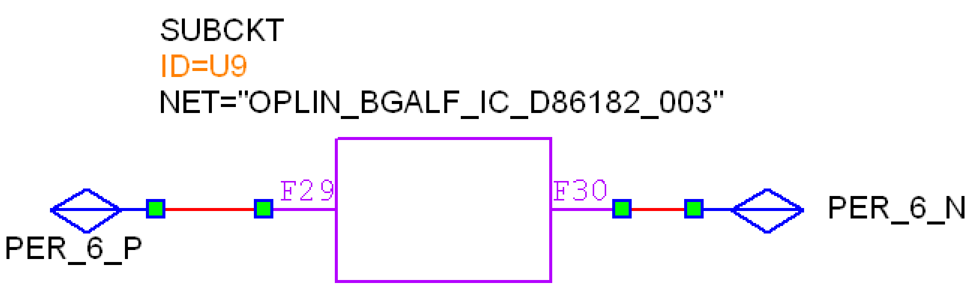
13.12.5. Using ACE

If you are using Automated Circuit Extraction (ACE), you should define the location of ground planes for conductor layers on the Element Options: STACKUP dialog box **Line Type** tab.

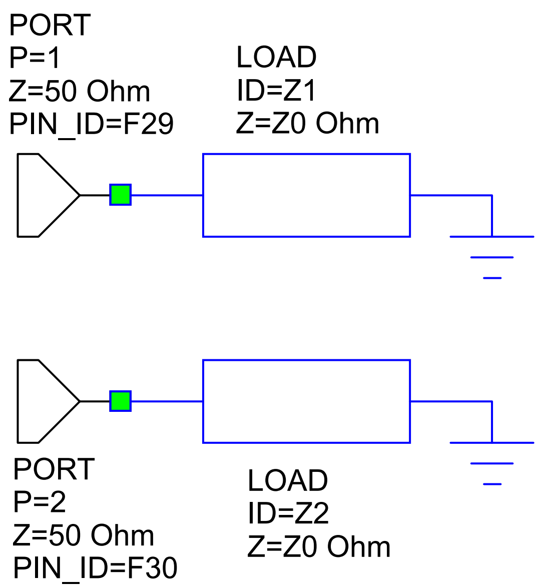


13.12.6. Schematic Components

The schematic created in the AWR Design Environment platform has the correct connectivity, but the actual components are unknown. To properly simulate the design you must add the component models to the subcircuits that are created. As a place holder, the model subcircuits simply contain PORT and LOAD elements. The following figure shows a schematic instance.

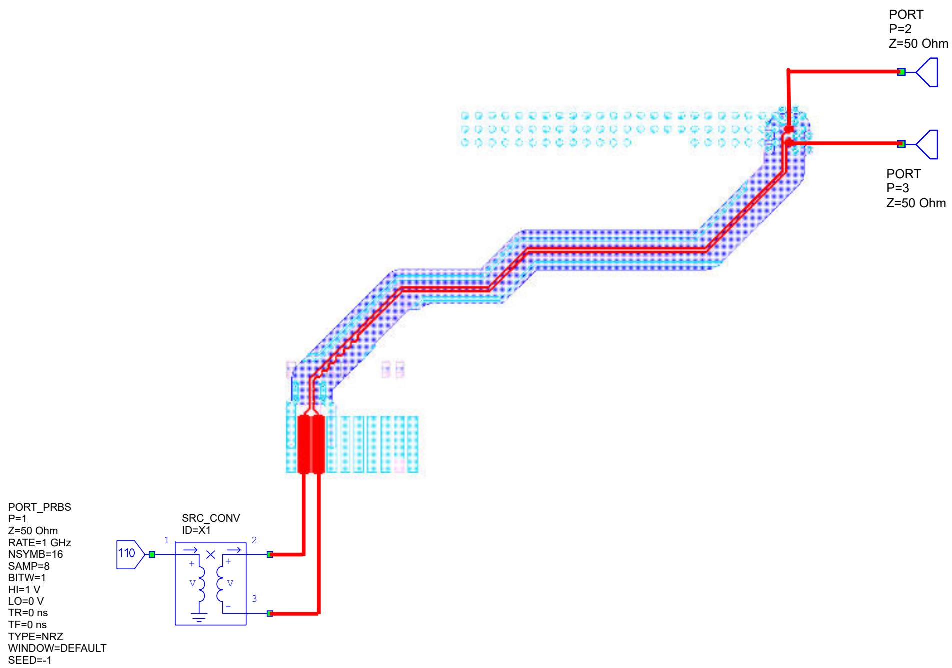


The following figure shows a default schematic instance model.

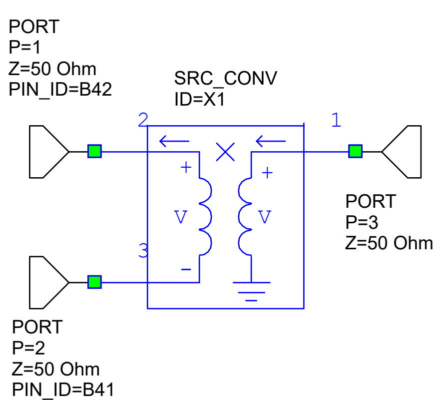


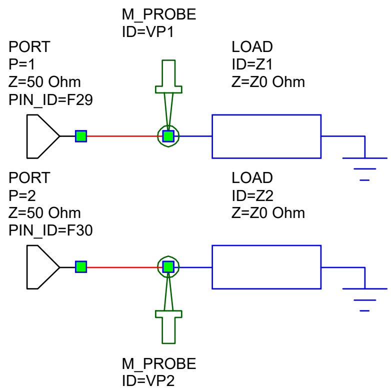
13.12.7. Adding Stimulus

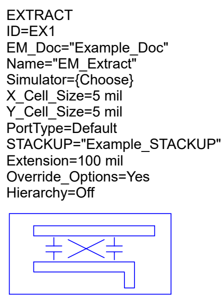
Commonly, you must add another node to the schematic subcircuit that represents the circuit stimulus (this is not necessary if the stimulus is a fully contained SPICE netlist file). The following figure shows a simple example circuit.

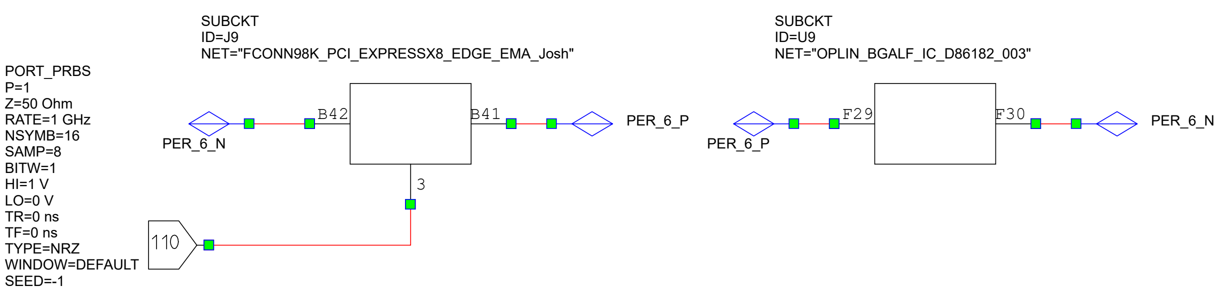


You can achieve this setup by modifying the top level subcircuits as shown in the following figures. Note that the third port added to the stimulus circuit appears as an additional pin on the top level schematic, which allows any desired voltage or current sources to be applied. The following figures show the stimulus circuit, receiver circuit, and the new top level schematic.







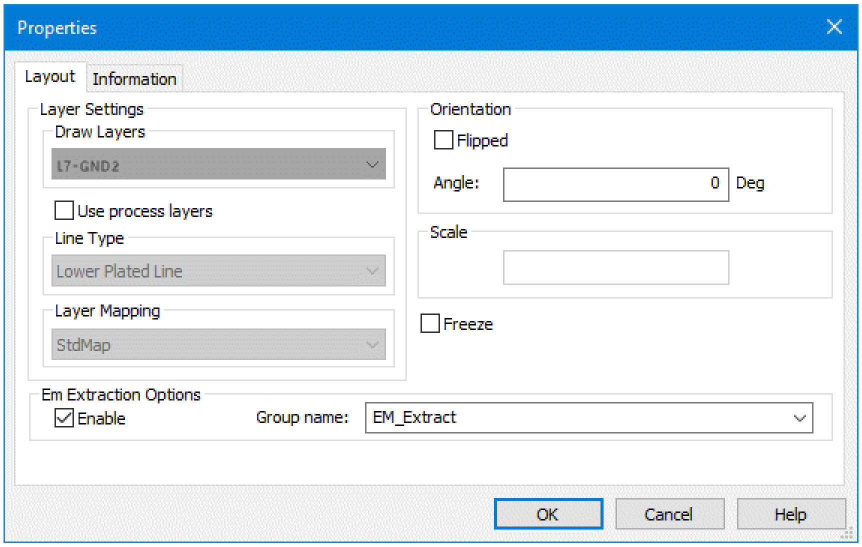


13.12.8. Extraction

The following sections include information about layout shapes and extraction ports.

13.12.8.1. Layout Only Shapes

You can use any available EM simulator to simulate the *.3Di* layout shapes imported by the PCB Import Wizard. By default, layout shapes associated with nets on the schematic (iNets) are sent to the EM document. To add other shapes to the EM simulation (such as ground planes) simply select the shapes in the layout, right-click and choose **Shape Properties**, select the **Enable** check box under **Em Extraction Options**, and ensure that the **Group name** matches the Name parameter on the EXTRACT block on the schematic (the default value after import is "EM_Extract").

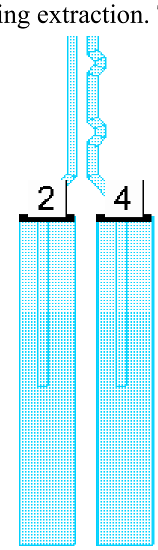


13.12.8.2. Ports

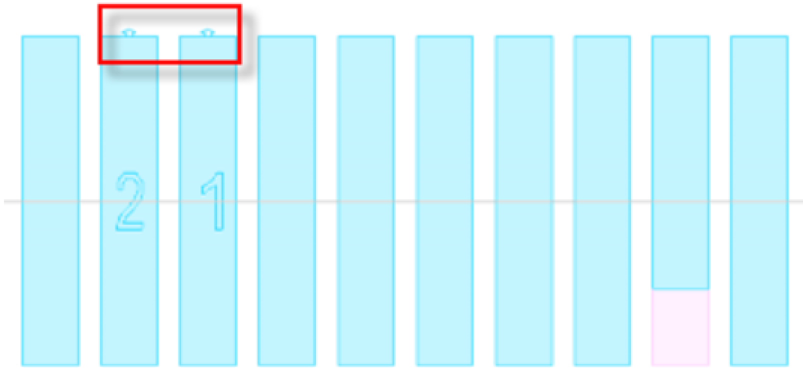
AWR Design Environment platform EM engines support a variety of port types. See "Extraction Ports" for information on setting up and selecting the appropriate extraction ports.

13.12.8.3. EM Pin Locations

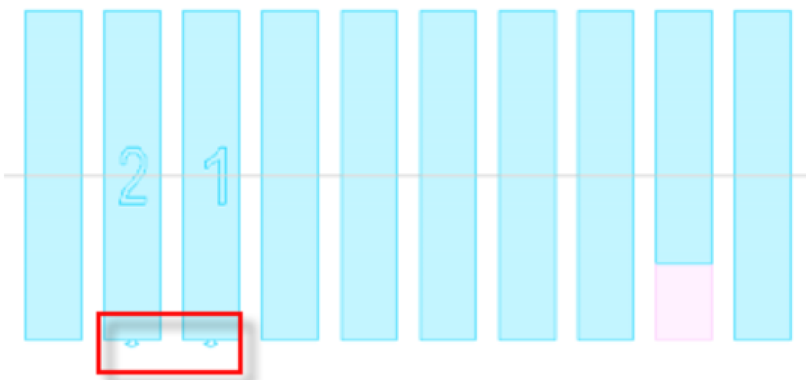
During extraction, ports are placed on the primary face of area pins. Because pins in PCB tools are points (usually in the center of the footprint geometry) the PCB Import Wizard does not have sufficient information to put the primary face anywhere but the first footprint geometry edge that is drawn. This can result in EM pins placed in less than ideal locations during extraction. The following figure shows the default EM pin placement.



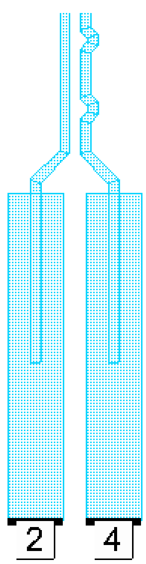
You can correct this by editing the appropriate component footprint and moving the primary face from the default location to the desired location. In the previous example, the primary face needs to be relocated from the "top" of the footprint geometry to the "bottom" of the footprint geometry. The following figure shows the original primary face location.



The following figure shows the modified primary face location.



As shown in the following figure, the extraction pins are now in the correct location.



13.12.9. Errors and Warnings

Errors and warnings from the PCB Import Wizard display in the Status Window. If the Status Window is not open, you should open it after importing to check the contents.

13.12.10. Solder Balls and Bumps

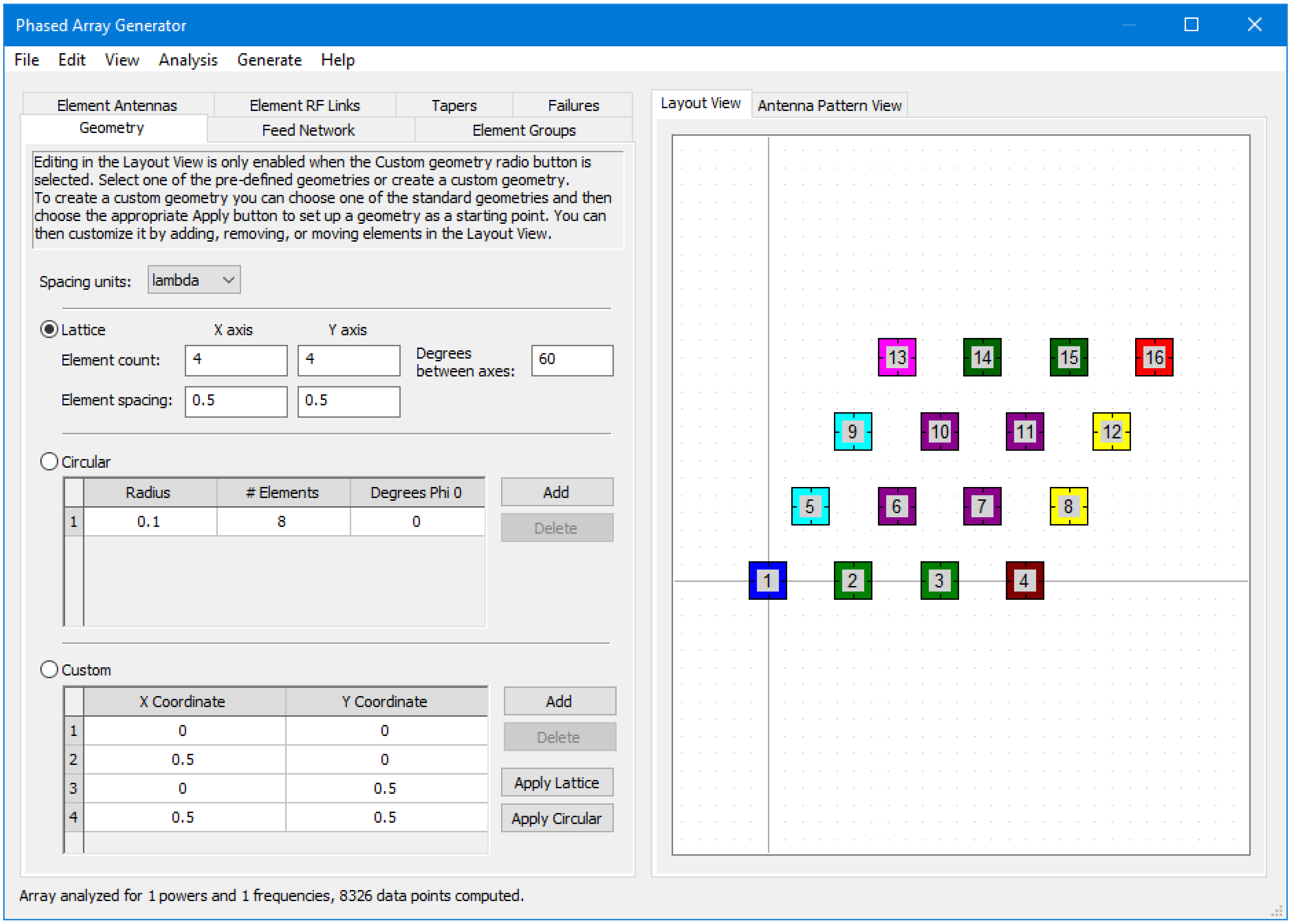
Contact Cadence Technical Support for more information about importing and simulating solder balls and bumps using the PCB Import Wizard.

13.13. Phased Array Generator Wizard

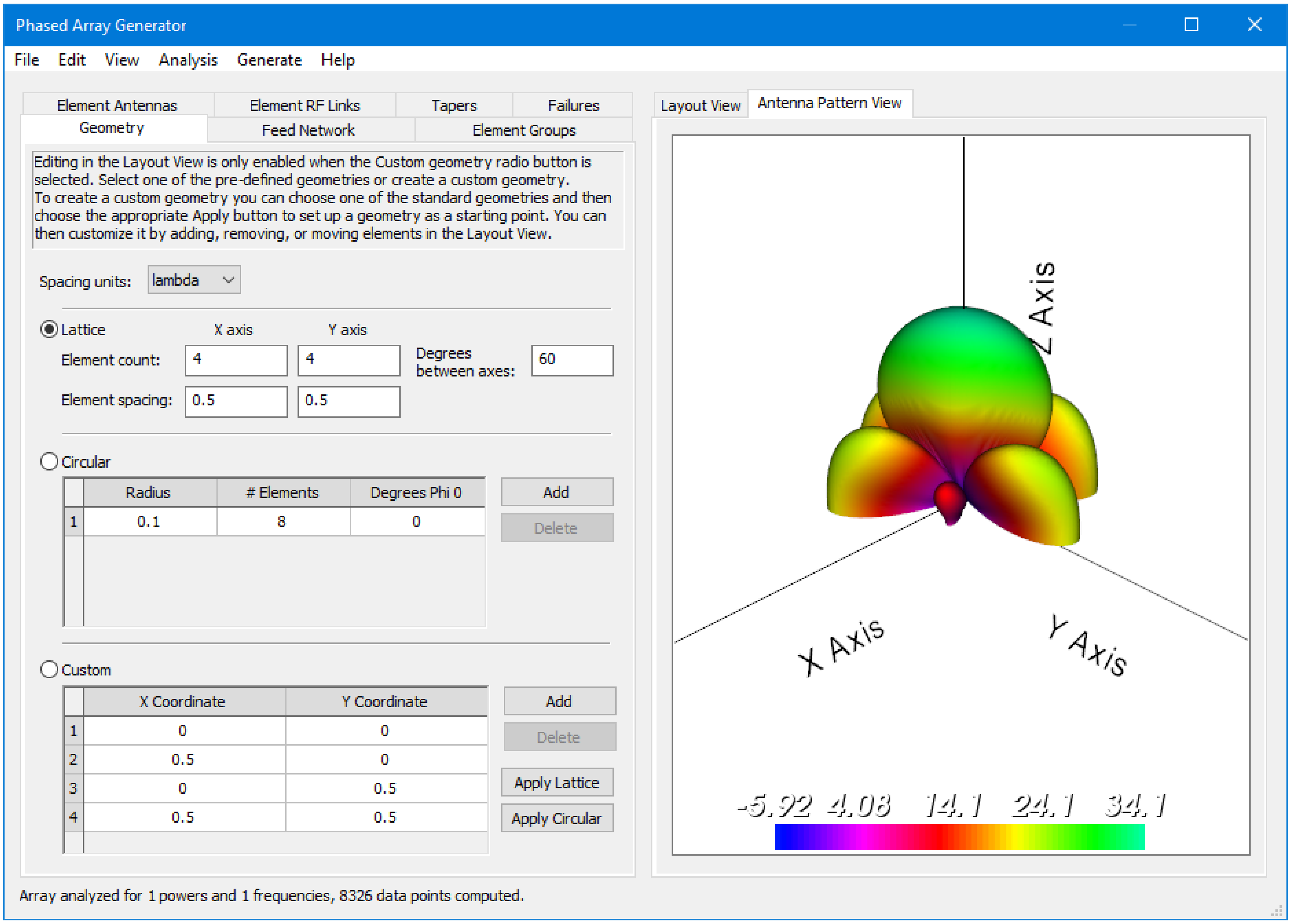
The Phased Array Generator wizard lets you interactively design a phased array antenna and then generate schematics or system diagrams that represent the design. With the wizard you can:

- Specify the 2D array geometry using either predefined lattice or circular arrangements or by specifying the x,y coordinates of individual elements.
- Specify basic characteristics of the feed network. When generating system diagrams, you can also choose to use a MIMO configuration rather than a phased array antenna configuration.
- Group elements together so they share the same characteristics.
- Specify the antenna characteristics, which are similar to the characteristics found in the ANTENNA block, if you are generating system diagrams.
- Specify an EM structure representing an individual antenna element if you are generating schematics.
- Specify settings for the RF links of each element group, such as gain and P1dB. Different settings may be specified for the transmit and receive configurations. When generating system diagrams, you may specify a Text Data File compatible with the AMP_F block's data file requirements in place of the behavioral settings such as gain and P1dB.
- Specify gain and phase tapers, choosing Dolph-Chebyshev, Taylor; or specify the tapers for individual elements.
- Specify element failures, either statistically or by selecting individual elements to fail.

A 2D layout of the antenna elements simplifies the task of selecting and visualizing the antenna elements. When using a custom geometry, you can drag individual elements to position them in the array.



To assist in the design process, a simulation of the overall gain of the array is performed automatically in the background and displayed as a 3D antenna pattern:



From the design you can generate:

- A set of system diagrams for simulation in Cadence Visual System Simulator™ (VSS) communications and radar systems design software. The various components of the array are implemented via subcircuits, which allows you to easily replace blocks as desired. For phased array antennas, a test bed system diagram that sweeps the antenna incidence angles and plots the gain in a graph may also be generated.
- A Text Data File compatible with the VSS PHARRAY_F block. For phased array antennas, a test bed system diagram that sweeps the antenna incidence angles and plots the gain in a graph may also be generated.
- A set of schematics and an EM structure for simulation with Cadence AXIEM® 3D planar EM analysis software. The EM structure models the physical antenna elements, incorporating the simulation of inter-element coupling effects. Note that the AXIEM-based model is limited to phased array antenna designs (no MIMO mode), and only models the transmit mode.

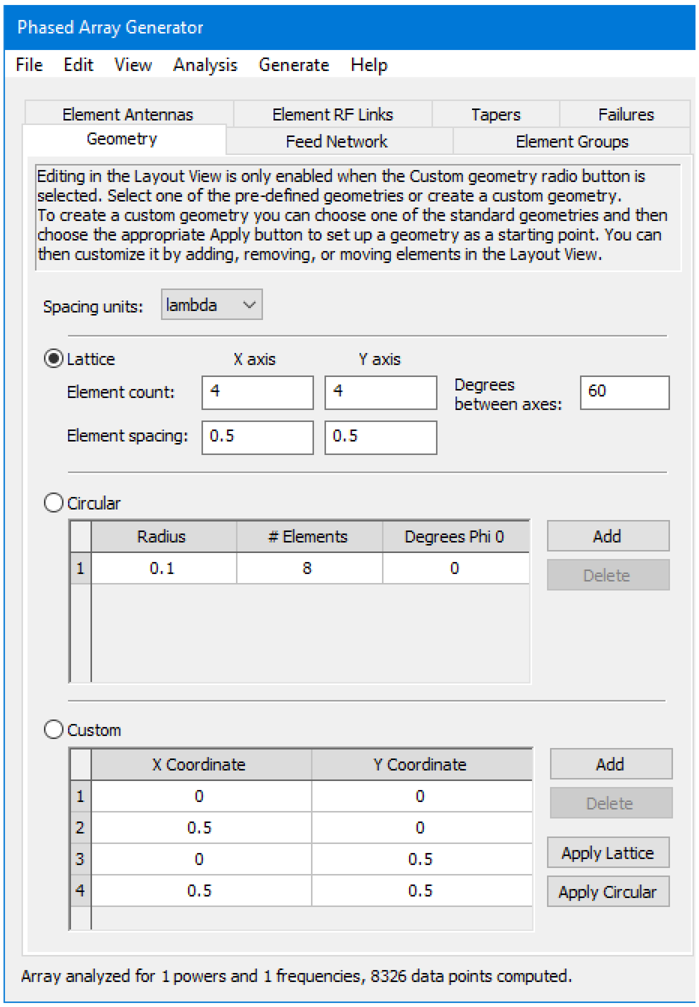
NOTE: The generation of system diagrams, PHARRAY_F data files, and schematics/EM structures requires either a VSS Radar Library (RDR-100) or a VSS 5G Library (W5G-100) license. The **Generate** menu options are only enabled if at least one of these licenses is available.

13.13.1. Designing an Array

The main generator window is divided into two halves. The left half contains a set of tabbed pages that are used to configure the array. The right half contains two tabs and is used to display either the Layout View or the Antenna Pattern View.

13.13.1.1. Geometry Tab

The **Geometry** tab is used to define both the number of elements and the locations of the elements. When you change the geometry, the Layout View representation auto-updates.



Spacing units determines the units represented by the spacing, radius, and coordinate field values. If "lambda" is selected, the units depend upon the design frequency. When generating system diagrams or schematics, the design frequency is specified when configuring the generation. For PHARRAY_F data files, the design frequency is determined by the PHARRAY_F block. For the Antenna Pattern View, the design frequency is the frequency represented. Note that this means changing the frequency of the Antenna Pattern View will have no effect on the antenna pattern displayed when "lambda" is chosen.

The **Lattice** geometry settings are used to specify elements arranged in evenly spaced horizontal rows, along the x-axis, with the elements in each row having the same spacing. The vertical alignment of the rows can be changed with the **Degrees between axes** setting.

The **Circular** geometry settings are used to specify elements arranged concentrically around the origin. Elements are specified by radial distance. For each radius, you specify the number of elements. The elements are distributed evenly on the circle at the specified radius. The location of the elements on that radius is controlled by the **Degrees Phi 0** setting which determines the angular location of the first element on the radius. The angle is measured in a counter-clockwise direction from the x axis.

The **Custom** geometry setting lets you specify the coordinates of individual elements. You can pre-populate the elements by clicking either the **Apply Lattice** or **Apply Circular** button. To use this feature, first enter either **Lattice** or **Circular** settings that approximate the desired geometry, then click the appropriate button. The Custom table contents are replaced with the same number of elements as in the chosen setting, and with the same coordinates.

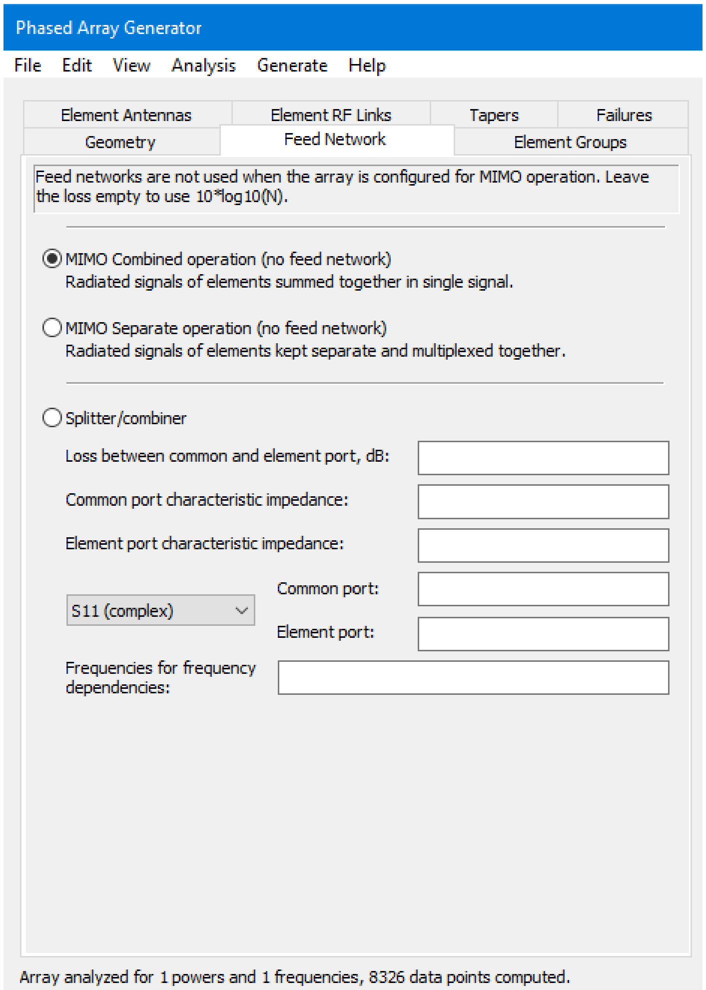
When **Custom** geometry is selected, you can drag elements in the Layout View to position them. You can also delete individual elements, which is helpful when creating a sparse array. For additional commands, such as aligning the selected elements, right-click in the Layout View.

The **Custom** geometry table supports pasting x,y coordinates from the Clipboard. The Clipboard data must consist of two columns. If you are copying the data from a text file, the columns should be separated by either tab characters or commas. If copying the data from a spreadsheet, select two and only two columns. In either case the data must consist of numeric values.

When pasting the Clipboard data into the **Custom** geometry table, how the geometry gets updated depends upon what is selected. If only a single cell is selected in the geometry table, elements represented by the data in the Clipboard are inserted into the existing geometry after the element containing the selected cell. If more than one cell is selected, the first and last selected elements and all elements in between are replaced by the elements represented by the data in the Clipboard.

13.13.1.2. Feed Network Tab

The **Feed Network** tab is used to defined how the elements are connected together on the circuit side.



There are two MIMO modes available. Note that schematic layout generation does not support the MIMO modes and generates a phased array configuration with a splitter based feed network. Also note that the Antenna Pattern View represents the antenna pattern of the array in the phased array configuration.

In the MIMO modes, the elements are treated as stand-alone elements and are not modeled with any RF circuit connection. For the **MIMO Combined operation** mode, the radiated signal represents the signal received at a point in space from all the elements. This is essentially the sum of the signal from each of the individual elements at a specified angle of incidence to the origin of the array.

For the **MIMO Separate operation**, the radiated signal represents the multiplexed separate signal from each individual element.

The **Splitter/combiner** setting selects the phased array configuration. The settings determine the characteristics of the feed network splitter/combiner. When generating system diagrams, you can specify either a single SPLITTER block for the entire feed network or a cascade of individual splitters.

The **Loss between common and element port**, dB setting determines the overall loss between the common port of the feed network and the port of an individual element. It is typically greater than or equal to the number of elements in dB. If left empty it is set to the number of elements in dB.

The characteristic impedance settings are set to _Z0 if left empty. _Z0 is the **Impedance** value specified on the System Simulator Options dialog box **RF Options** tab.

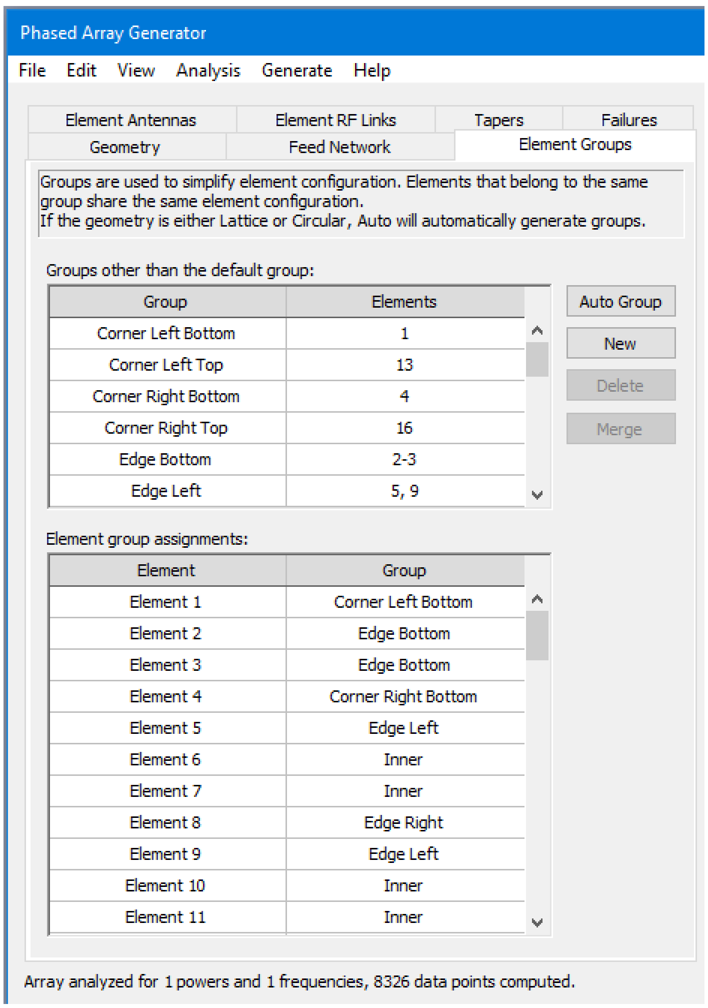
The **S11**/**Return Loss**/**VSWR** settings, if left empty are set to the equivalent of S11 = 0.

Frequency-dependent settings may be specified for any of the above splitter settings by entering an array containing the frequency-dependent values, and then entering an array containing the frequencies corresponding to those values in **Frequencies for frequency dependencies**.

Note that frequency-dependent settings are not currently supported when generating system diagrams or schematic layouts. They are also not modeled by the Antenna Pattern View.

13.13.1.3. Element Groups Tab

The **Element Groups** tab is used to manage element grouping. Element groups allow you to manage antenna and RF link configurations for groups of elements. Element groups also help organize the Text Data files generated for PHARRAY_F blocks.



An element belongs to either the [Default] group or a named group. An element can only belong to one group at a time. All elements initially belong to the [Default] group.

With the exception of the elements in the [Default] group, all the elements in a group share the same antenna configuration and the same RF link configuration. Elements in the default group by default do share the same antenna and RF link configurations, which are the [Default] antenna configuration and the [Default] RF Link configuration, but the elements may also be assigned different antenna or RF link configurations.

There are several ways of creating and assigning elements to a group. The simplest method, if the array geometry is either **Lattice** or **Circular**, is to click the **Auto Group** button. This option is not available for **Custom** geometries.

For **Lattice** geometries, clicking **Auto Group** generates up to nine groups, depending on the number of elements in each row and column. The groups generated are:

- One group for each corner element (up to four groups).
- One group for the elements along the left, top, right, and bottom edges of the array, excluding the corner elements (up to four groups).
- One group containing all the remaining elements (one group).

For **Circular** geometries, clicking **Auto Group** generates a group for each radius entry.

Another means of specifying an element group is to select in the Layout View elements to be grouped together. Right-click and choose **Change Group > Create Group** to create a new group, or choose one of the existing groups from **Change Group** menu.

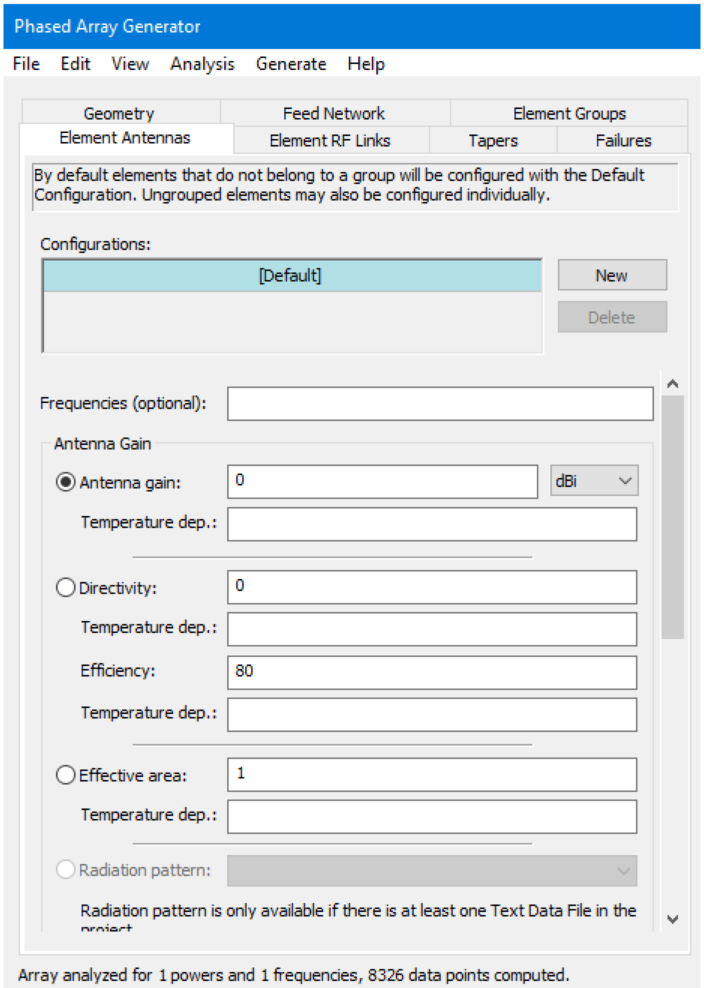
The third means of changing an element's group is to select the desired element in the **Element group** assignments table, then double-click on the group. A drop-down list of the available groups displays, allowing you to change the group.

Some additional features to note:

- Selecting a group in the **Groups other than the default group** table selects all the elements of that group in the Layout View and the **Element group** assignments table.
- You can right-click in the Layout View and choose **Select Group** to select the elements of a group by location.
- With an element selected in the Layout View, you can right-click and choose **Extend Selection to All Elements in Group** to select all the elements of the group that contains the selected element.
- Groups are color coded in the Layout View by default. All elements belonging to a group are surrounded with a border of the same color, provided the individual elements in the view are displayed large enough. You can toggle on/off the group coloring by choosing **View > Display > Element Groups**.

13.13.1.4. Element Antennas Tab

The **Element Antennas** tab is used to specify the antenna characteristics of the array elements. Note that when generating schematic layouts, the antenna settings are ignored, as the antenna characteristics are determined by the EM structure.



Antenna characteristics are defined via antenna configurations. With the exception of the [Default] group, all elements in a group share the same antenna configuration. In the [Default] group individual elements may be assigned to individual antenna configurations. By default all groups are assigned the [Default] antenna configuration.

Each set of antenna characteristics is defined as an antenna configuration. Antenna configurations are defined by clicking the **New** button to add new antenna configuration to the configuration list. You can change the name of the new configuration by double-clicking the name. Configuration names may only contain the letters of the alphabet, the digits 0 through 9, and the underscore and space characters.

You can also create an antenna configuration by selecting one or more element groups in the Layout View, right-clicking and choosing **Assign Antenna Configuration > Create Configuration**.

The settings for an antenna configuration are similar to the antenna settings of the RF ANTENNA block.

Note that in order to specify a radiation pattern Text Data file, the Text Data file must already be present in the Project Browser **Data Files** node.

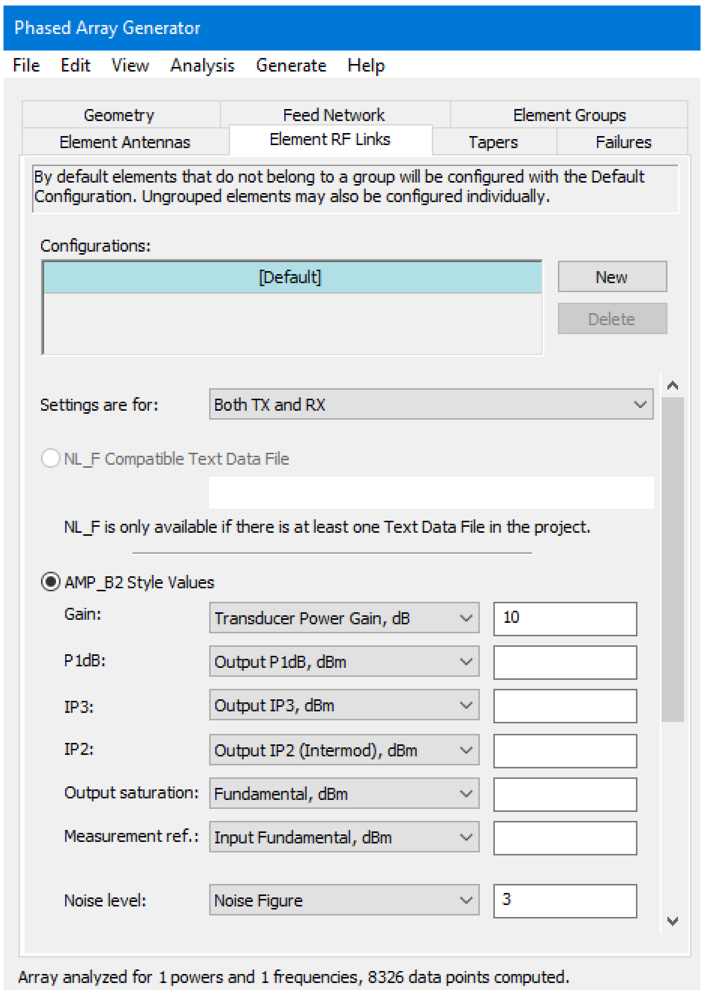
You assign an array characteristic to a group or an individual element belonging to the [Default] group by selecting the group/element in the Layout View, right-clicking and choosing **Assign Antenna Configuration** and the desired antenna configuration.

Some additional features to note:

- Selecting an antenna configuration in the **Configurations** table selects all the elements in the Layout View assigned that antenna configuration.
- You can select the elements assigned an antenna configuration by right-clicking in the Layout View, choosing **Select Antenna Configuration** and choosing an antenna configuration.
- Selecting an individual element in the Layout View that belongs to a group other than the [Default] group selects all the elements in the same group as the selected element.
- You can select an element or group in the Layout View, right-click, and choose **Extend Selection to All with Antenna Configuration** to select all the elements with the chosen antenna configuration.

13.13.1.5. Element RF Links Tab

The **Element RF Links** tab is used to specify the RF link characteristics of the array elements.



Specifying RF link characteristics is similar to specifying antenna characteristics on the **Element Antennas** tab. RF link characteristics are defined via RF link configurations. With the exception of the [Default] group, all elements in a group share the same RF link configuration. In the [Default] group individual elements may be assigned to individual RF link configurations. By default all groups are assigned the [Default] RF link configuration.

Each set of RF link characteristics is defined as an RF link configuration. To define a new link configuration and add it to the configuration list, click the **New** button. You can change the name of the new configuration by double-clicking on the name. Note that configuration names may only contain the letters of the alphabet, the digits 0 through 9, and the underscore and space characters. You can also create an RF link configuration by selecting one or more element groups in the Layout View, right-clicking and then choosing **Assign RF Link Configuration > Create Configuration**.

An RF link configuration may have two different sets of settings: one for when the array is transmitting and one for when the array is receiving, or it may be configured with a single set of settings used for both transmit and receive. The specified set is determined by the **Settings are for** selection.

RF link characteristics may either be specified as a Text Data file to be used by the Frequency Dependent Behavioral Amplifier (File-Based) block (AMP) or via amplifier characteristics such as gain and P1dB similar to those of the Behavioral Amplifier, 2nd Generation block (AMP_B2). Note that AMP_F Text Data files are not supported when generating schematic layouts, as the Microwave Office program does not have a compatible AMP_B2 element.

AMP_F Text Data files is only available if there is at least one Text Data file present in the Project Browser **Data Files** node.

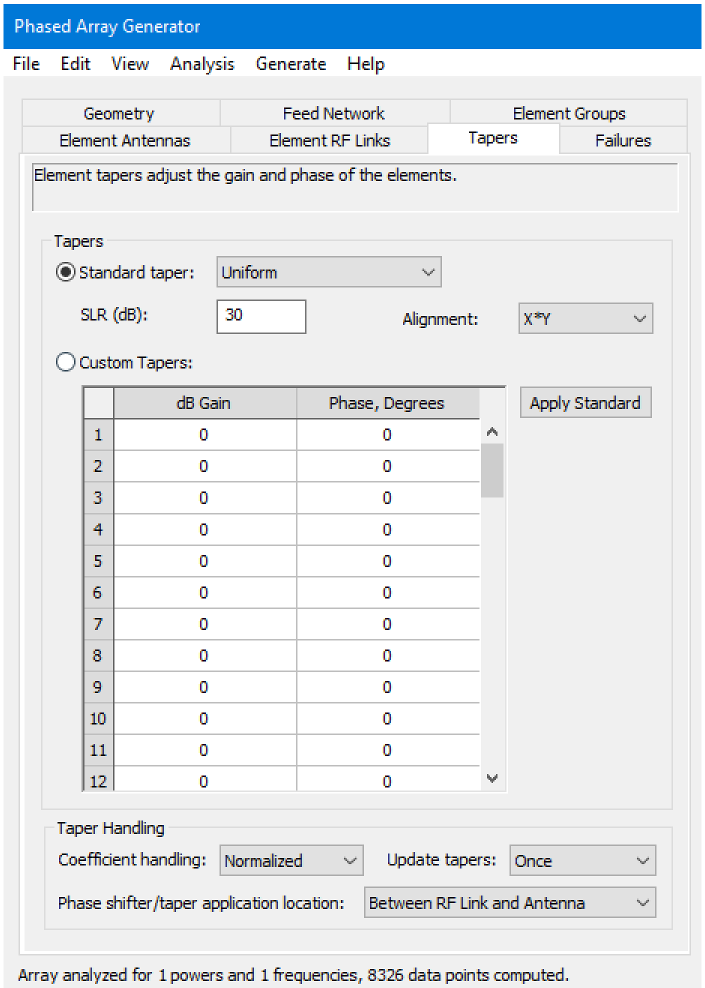
An RF link configuration is assigned to a group or an individual element belonging to the [Default] group by first selecting the group/element of interest in the Layout View. Right-click in the Layout View and choose **Assign RF Link Configuration** and then the desired RF link configuration option.

Some additional features to note:

- Selecting an RF link configuration in the **Configurations** table selects all the elements in the Layout View assigned to that RF link configuration.
- You can right-click in the Layout View and select the elements assigned an RF link configuration by choosing **Select RF Link Configuration** and then the RF link configuration option.
- Selecting an individual element in the Layout View that belongs to a group other than the [Default] group selects all the elements belonging to the same group as the selected element.
- If you have an element or group selected in the Layout View, you can right-click and choose **Extend Selection to All with RF Link Configuration** to select all the elements with the chosen RF link configuration.

13.13.1.6. Tapers Tab

The **Tapers** tab is used to assign gain and phase tapers to the elements of the array.



The following algorithmically defined tapers may be applied via **Standard taper** options:

- **Uniform**: No taper is applied.
- **Dolph-Chebyshev** tapering is applied. The taper is calculated using the Dolph-Chebyshev technique and is configured by defining the **SLR** (side lobe ratio) in dB. If the array geometry is **Lattice**, you can define the taper **Alignment** (whether the taper is calculated along the X axis, Y axis, or as a multiplication of tapers calculated along each axis).
- **Taylor** tapering is applied. The **SLR** and **Alignment** settings apply similar to the Dolph-Chebyshev tapering.

Custom tapering may also be applied by specifying the gain and phase values for individual elements. Click the **Apply Standard** button to initialize the gain and phase tapers in the **Custom Tapers** table based upon the selected **Standard taper**.

The **Custom Tapers** table supports pasting gain and phase values from the Clipboard. The following describes the process:

- The data in the Clipboard may have either one or two columns, depending upon the left-most and top-most selected cells in the **Custom Tapers** table.
- If the left-most selected cell is in the gain column, the Clipboard data may have either one or two columns. The first column is the gain data. The second column if present is the phase data.

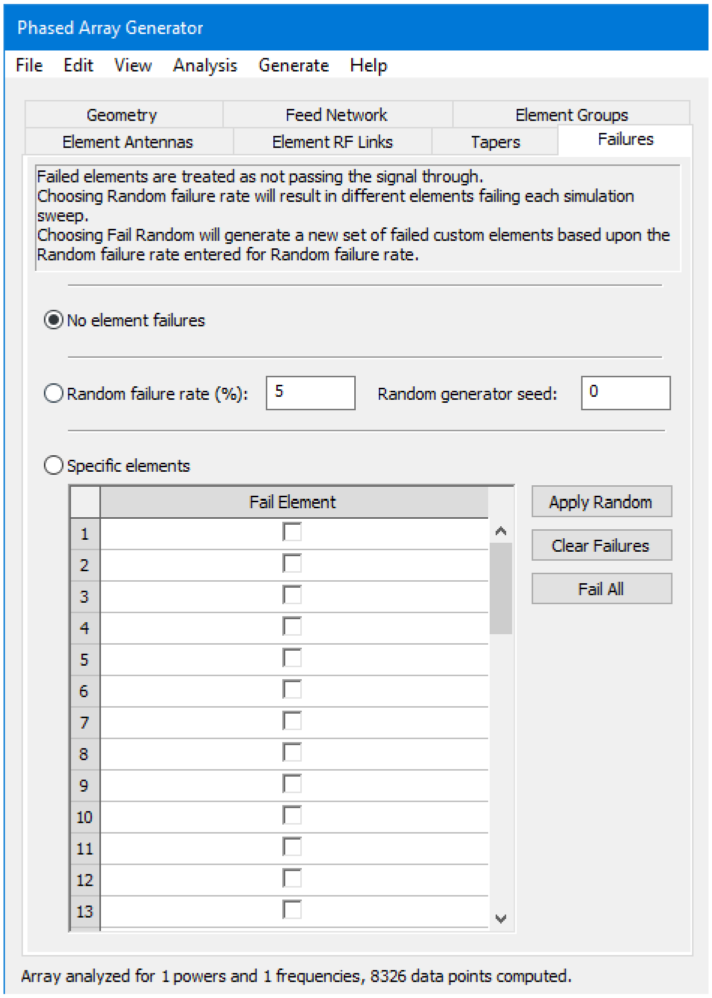
- If the left-most selected cell is in the phase column, the Clipboard data may only have one column. This column is the phase data.
- The data from the Clipboard is pasted starting at the left-most selected column and the top-most selected row, and replaces any overlapped data.
- The Clipboard data must 'fit' based upon the left-most selected column and the top-most selected row. For example, if there are 16 elements, and the tenth row is selected, the Clipboard data uses from 1 to 6 rows of data. If there are 7 or more rows of data, the **Paste** command is not available.
- If obtaining the Clipboard data from a text file, the columns should be separated by either tab characters or commas. The data may also be copied from a spreadsheet; in that case simply select the corresponding number of rows and columns to copy to the Clipboard. Note that all cells must contain a numeric value; cells may not be empty.

Some additional features to note:

- The magnitude of the taper gains are indicated by the color of the inner band of the elements in the Layout View when the element size is large enough. The more negative the dB gain, the more red, the more positive the dB gain the more blue, 0 dB displays as white. You can toggle on/off the gain taper coloring by choosing **View > Display > Gain Tapers**.
- The angle of the taper phases are also indicated by the color of the inner band of the elements in the Layout View when the element size is large enough. The more negative the angle, the more yellow, the more positive the angle the more aqua, 0 degrees displays as white. You can toggle on/off the phase taper coloring by choosing **View > Display > Phase Tapers**.
- The **Coefficient handling** and **Update tapers** settings are used by the PHARRAY_F block.
- The tapers are applied at the same point as the steering phase shift. The **Phase shifter/taper application location** setting determines where the phase shifter and gain tapers are applied. For transmit configurations, applying the gain tapers **Between Feed Network and RF Link** causes the array antenna pattern shape to change as the amplifiers start to compress, while applying the gain tapers **Between RF Link and Antenna** does not significantly modify the shape of the array antenna pattern, only the gain levels. The reverse is true for receive configurations.

13.13.1.7. Failures Tab

The **Failures** tab is used to mark individual elements as 'failed'. A failed element is similar to an element whose RF link has a high loss (very negative dB gain).



Element failures may be generated pseudo-randomly by selecting **Random failure rate**. In a given array, any given **Random generator seed** value results in the same elements failing. When working with the PHARRAY_F block, these settings have additional behaviors. Setting a negative value for the **Random failure rate** causes the block to generate a new set of failed elements each sweep, while setting the **Random generator seed** value to 0 causes the block to use a random generator seed based upon the ID parameter of the block. See PHARRAY_F for more information.

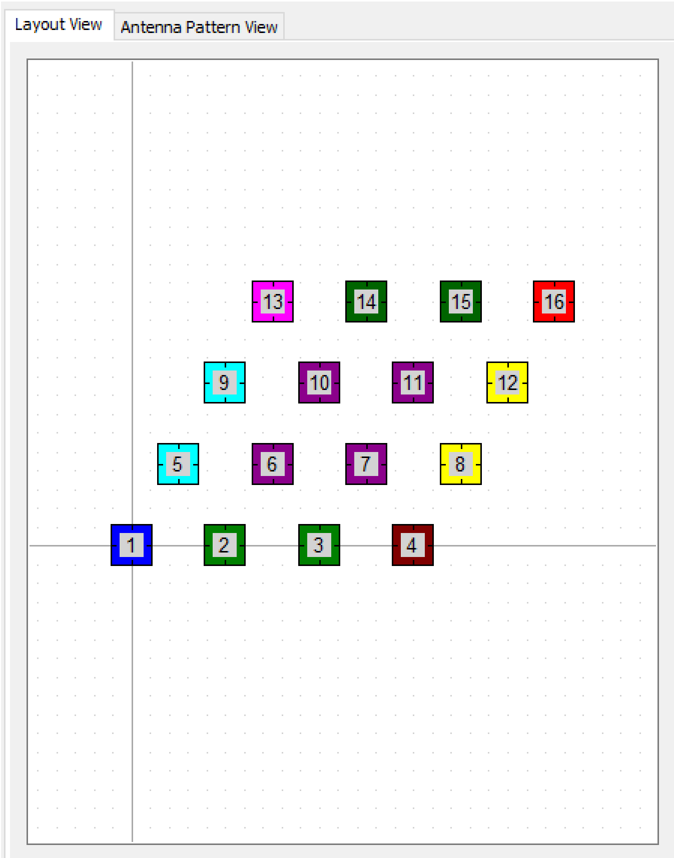
You can also specify individual elements for failure by selecting **Specific elements** and clicking on an element to toggle the failure state of the element. If multiple elements are selected, the failure state of each selected element is toggled.

Some additional features to note:

- Failed elements by default are marked with an 'X' when the element size is large enough. You can toggle this on/off by choosing **View > Display > Element Failures**.
- When one or more elements are selected, you can right-click in the Layout View and choose an option to either toggle the failure state of the selected elements, mark all the selected elements as failed, or mark all the selected elements as not failed.

13.13.1.8. Layout View

The Layout View presents a 2D view of the array elements.



Depending on the current zoom level, the following may be indicated in the element's display:

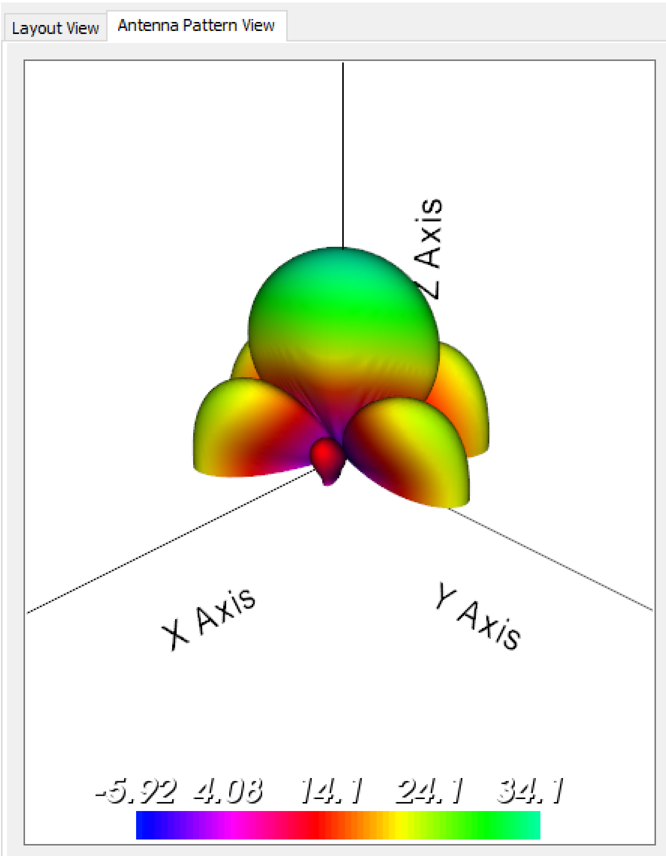
- The id number of the element; the first element is number 1.
- The group the element belongs to, indicated by the color of the outer band. Each element group has a unique color.
- The gain taper of the element, indicated by the color of the inner band surrounding the element id. The more negative the gain in dB the deeper red the color, the more positive the gain in dB the deeper blue the color, with 0 dB gain tapers represented by white.
- The phase taper of the element, indicated by the color of the inner band surrounding the element id. The more negative the phase angle the more yellow, the more positive the phase angle the more aqua, with 0 phase represented by white.
- The failure state of the element. Failed elements are marked with an 'X'.

You can toggle the view of these indications individually by choosing **View > Display** and the appropriate option.

The Layout View supports additional behaviors for the **Geometry**, **Element Groups**, **Element Antennas**, **Element RF Links**, and **Failures** tabs.

13.13.1.9. Antenna Pattern View

The Antenna Pattern View presents a 3D polar view of the overall array power gain expressed in dB. The 3D surface displayed indicates the strength of the gain by both the distance of the surface point from the origin and by color. The color scale in the view indicates the dB values represented by the different colors.



The gains displayed are computed automatically in a background analysis of the array design that is updated each time the array settings are modified. The current status of the analysis is displayed at the bottom of the main generator window.

Note that the analysis is a simplified analysis and you should treat the gain values displayed subjectively. A simplified saturation model is used for modeling amplifier compression/saturation, and element coupling effects are not modeled. A full VSS or EM simulation should be performed to obtain quantitative gain values.

Choose **Analysis > New Floating Antenna Pattern View** to open an Antenna Pattern View in a floating window. You can open multiple windows. Floating Antenna Pattern View windows let you view the 3D antenna pattern from different orientations and are helpful for visualizing the effects of changes made within the Layout View, such as dragging an element in a custom geometry or toggling an element's failure state when using specific element failures.

When an Antenna Pattern View is active, an Analysis floating window displays. This window contains the following tunable settings:

- **Frequency** - the signal frequency the background analysis uses. Note that if **Spacing units** on the **Geometry** tab is set to "lambda" you do not see any effect when you change the frequency, as the spacing between elements is directly related to the frequency.

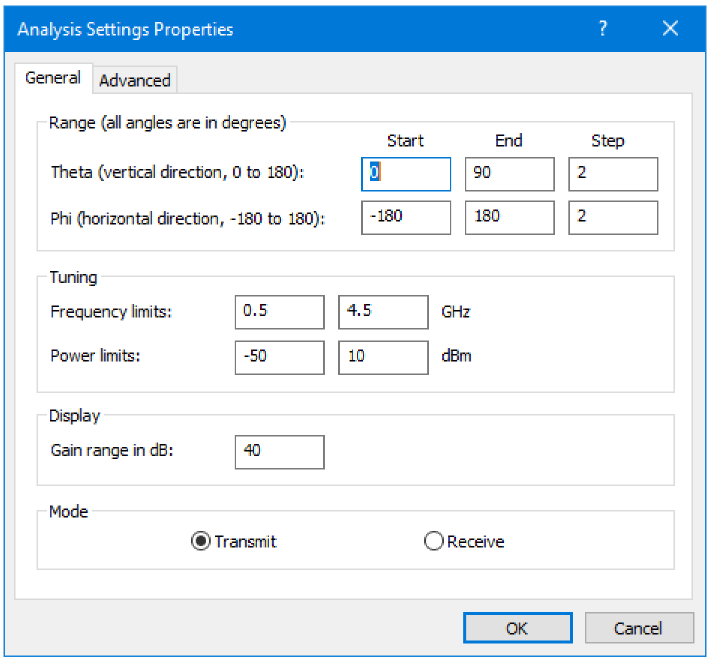
- **Power** - the power of the signal at the common port of the feed network.
- **Steering Theta** - the theta steering angle in degrees. A theta of 0 degrees is along the z-axis.
- **Steering Phi** - the phi steering angle in degrees. Positive angles rotate the positive x-axis towards the positive y-axis around the z-axis.

To avoid flicker in the view, press the **Shift** key while dragging one of the tuner bars to prevent the antenna pattern view from updating until the key is released or the dragging is completed.

NOTE: To re-open the Analysis floating window, first select the Layout View tab to display the Layout View, then select the Antenna Pattern View tab.

The steering angles are used in the analysis to compute the phase shift values for the element phase shifters.

Choose **Analysis > Analysis Settings** to display the Analysis Settings Properties dialog box to configure settings used for the background analysis and the antenna pattern view.



The following settings are available:

- Range of angles analyzed - these settings are used to define the theta and phi incident angles that are analyzed. Theta is measured from the z-axis. When phi is 0, positive theta angles represent a rotation of the positive z-axis about the y-axis towards the positive x-axis. Positive phi angles represent a rotation of the positive x-axis about the z-axis towards the positive y-axis. The limits of the **Steering Theta** and **Steering Phi** settings in the Analysis floating window are the same as the limits of the corresponding range setting.
- Frequency limits - these values define the range of the **Frequency** setting in the Analysis floating window.
- Power limits - these values define the range of the **Power** setting in the Analysis floating window.
- Gain range in dB - this value is used to define the lower dB gain value displayed, and is relative to the maximum dB gain. The maximum dB gain from the analysis minus the gain range setting is the minimum dB gain value displayed, and maps to the origin. Setting a smaller value for the gain range has the effect of increasing the resolution of the gains displayed.
- Mode - this value determines which direction the analysis is performed, either transmit or receive.

- Enable multi-core processing - you can use this setting to disable the use of multiple processor cores should issues occur when displaying the Antenna Pattern View. Disabling multiple processor cores typically slows down updating of the 3D view whenever an array setting is changed.

NOTE: Should the Antenna Pattern View not reflect an expected pattern, try changing one of the settings in the Analysis floating window and then changing it back to force a recalculation of the analysis and display.

13.13.2. Generating System Diagrams and Schematics

You use the **Generate** menu to select what is to be generated.

NOTE: The generation of system diagrams, PHARRAY_F data files, and schematics/EM structures requires either a VSS Radar Library (RAD-100) or a VSS 5G Library (W5G-100) license. The **Generate** menu options are only enabled if at least one of these licenses is available.

13.13.2.1. Generate System Diagrams

This option generates a set of system diagrams representing the array design. It offers the following:

- A system diagram representing the full array design. This system diagram has an input port and an output port and can be used as a DUT subcircuit.
- A Text Data file is generated containing the coordinates and gain and phase tapers of each element.
- A system diagram subcircuit representing the feed network for phased array configurations or the multiplexer/demultiplexer for MIMO configurations is created.
- For phased array configurations, the feed network may either be composed of a single SPLITTER block, or be composed of cascaded splitter subcircuits. For the cascaded splitters, splitter subcircuits are generated from combinations of elemental 2- and 3-way SPLITTER blocks.
- A system diagram subcircuit representing each element group is created. An instance of the appropriate subcircuit is included in the DUT subcircuit for each element.
- A system diagram subcircuit representing each RF link configuration is created. Each element group subcircuit includes an instance of the applicable RF link configuration subcircuit.
- A system diagram subcircuit representing each antenna configuration is created. Each element group subcircuit includes an instance of the applicable antenna configuration subcircuit.
- A system diagram subcircuit representing the element phase shifter is created. This subcircuit is used by all the elements. This subcircuit is also responsible for applying the gain and phase taper for each element. This subcircuit handles computing the appropriate phase shift for a given pair of steering angles for the element at its particular coordinates. Each element group subcircuit includes an instance of this phase shifter subcircuit.
- Impedance mismatch modeling is enabled for the project.
- If the array design is a phased array configuration and not MIMO, the option of generating a full test bed is offered. The test bed consists of the following:
  - A system diagram containing the DUT subcircuit along with an optional SWPVAR block for sweeping the incident theta angle and an optional SWPVAR block for sweeping the incident phi angle. The test bed system diagram contains a PORT_SRC for the source signal and a termination PORT.
  - A graph with a Cascaded Gain (C_GP)measurement configured to display the swept gain.

13.13.2.2. Generate PHARRAY_F Data File

This option generates a Text Data file that the Phased Array Assembly, Data-file based block (PHARRAY_F) can use.

If the array design is a phased array configuration and not MIMO, the option of generating a test bed is offered. The test bed consists of the following:

- A system diagram containing a PHARRAY_F block with the data file set to the generated Text Data file, along with an optional SWPVAR block for sweeping the incident theta angle and an optional SWPVAR block for sweeping the incident phi angle. The test bed system diagram contains a PORT_SRC for the source signal and a termination PORT.
- A graph with a Cascaded Gain (C_GP) measurement configured to display the swept gain.

13.13.2.3. Generate Schematic Layout

This option generates a set of schematics and an EM structure representing the array design. The RF link, phase shifters, and feed network are represented by the schematics, while the antenna characteristics are represented by an EM structure.

Note that this option does not support the MIMO configurations. The antenna configurations are also ignored, as the antenna properties are defined by the EM structure.

The following schematics are generated:

- A master schematic representing the full array design.
- A schematic subcircuit representing the feed network. For more than three elements, the feed network is made up from a combination of smaller feed network subcircuits, down to SPLIT2 and SPLIT3 elements.
- A schematic subcircuit representing each element group is created. An instance of the appropriate subcircuit is included in the master schematic for each element.
- A schematic subcircuit representing each RF link configuration is created. Each element group subcircuit includes an instance of the applicable RF link configuration subcircuit.
- A schematic subcircuit representing each antenna configuration is created. Each element group subcircuit includes an instance of the applicable antenna configuration subcircuit. This subcircuit only contains a GPROBE2 element and does not apply antenna properties, as the antenna properties are determined by the EM structure.
- A schematic subcircuit representing the element phase shifter is created. This subcircuit is used by all the elements and is also responsible for applying the gain and phase taper for each element. This subcircuit handles computing the appropriate phase shift for a given pair of steering angles for the element at its particular coordinates. Each element group subcircuit includes an instance of this phase shifter subcircuit.

The EM structure representing the array of antennas can be generated by either selecting an EM structure representing an individual antenna and then having it duplicated at the appropriate element locations, or by specifying settings for a simple rectangular patch antenna that is then duplicated at the appropriate element locations.

Cadence recommends that you create an EM structure representing an individual antenna element and have the generator copy that structure. This lets you set the various EM structure settings as needed for the antenna design. The EM structure must be in the project in order for you to select it when generating a schematic layout. Only EM structures with at least one EM port and all EM ports having the same port number are supported.

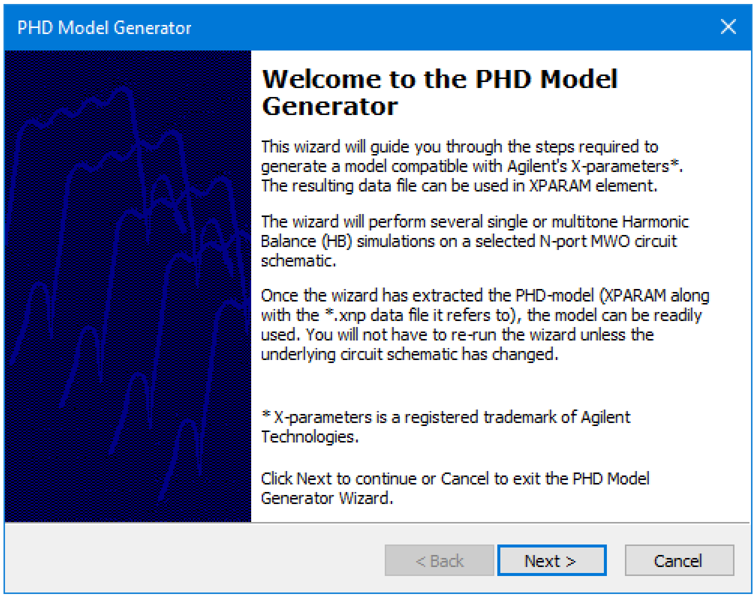
An ANTENNA_CKT_3D EM annotation is added to the generated EM structure. The results of the simulation can then be viewed by selecting the 3D EM view for the generated EM structure.

Note that when the schematics and EM structure are generated, the Status Window may display an error similar to: "At least one frequency must be specified to simulate." You can ignore this error.

13.14. PHD Model Generator Wizard

The PHD Model Generator Wizard creates a Poly-Harmonic Distortion (PHD) model (compatible with Agilent's X-parameters®) from an Microwave Office software circuit with one or more ports. You can use the resulting data file in an XPARAM element.

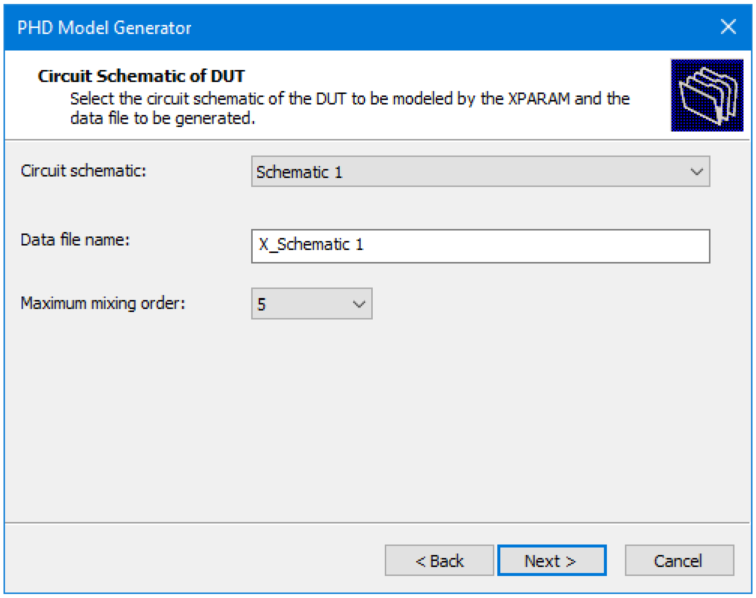
To access the PHD Model Generator Wizard, open the **Wizards** node in the Project Browser and double-click **PHD Model Generator**. The PHD Model Generator dialog box displays as shown in the following figure.



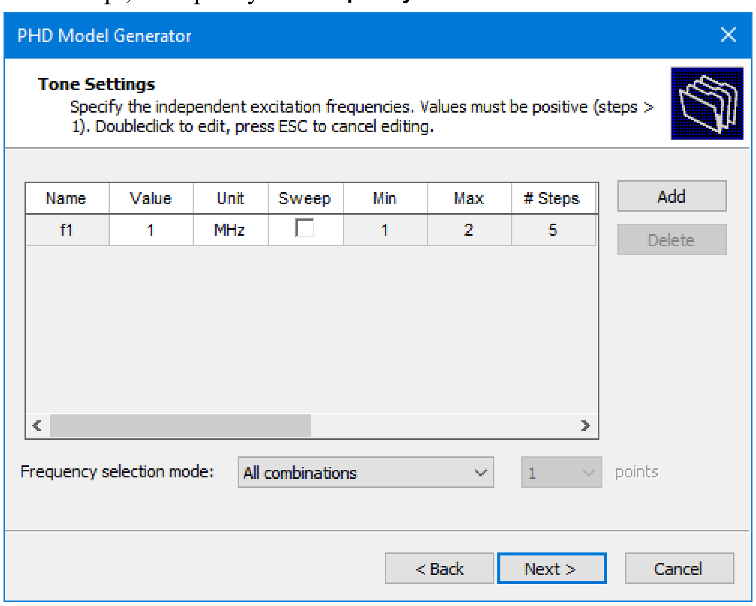
To use the PHD Model Generator Wizard:

1. Click **Next** to proceed with the wizard.

   Select the **Circuit schematic** to be modeled by the XPARAM, the **Data file name** of the file you want to generate, and the **Maximum mixing order** value. The mixing order does not determine the harmonic component selection of Harmonic Balance analysis. If single-tone HB analysis is done with 10 harmonics and maximum mixing order is 5, the results of 5 harmonics are stored in the file.



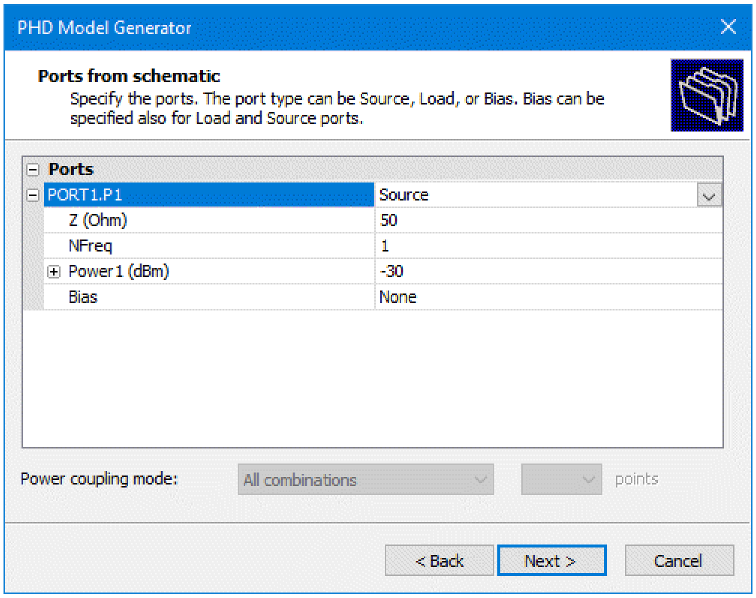
2. Specify the excitation frequencies (tone settings) for the HB analysis, select **Sweep** if the fundamental frequency is to be swept, and specify the **Frequency selection mode**.



The **Frequency selection mode** determines how the frequencies are swept if there are several tones.

- **All combinations** results in a larger set of frequencies, where all the combinations of swept frequencies are included. If f1 is swept from 1 to 3 with 3 points, and f2 is swept from 10 to 30 with 3 points, the resulting frequencies are: [1,10], [1,20], [1,30], [2,10], [2,20], [2,30], [3,10], [3,20], and [3,30].
- **Coupled** results in a frequency set where each tone is swept independently. All the tones share the number of frequency points specified in **points**. If f1 is swept from 1 to 3 and f2 is swept from 10 to 30 with 3 points, the resulting frequencies are: [1,10], [2,20], and [3,30].

3. Define the port type in the Microwave Office schematic as **Source**, **Load**, or **Bias**.



If the port type is **Source**:

- Z (ohm) specifies the port impedance.
- NFreq defines how many fundamental frequencies the port excites.
- PowerN defines the port power at frequency N. There are NFreq power definitions in total (Power1, Power2, …, PowerN).
  - If **Sweep** is selected, the power is defined by **Min**, **Max**, and **Steps**. **Angle (Deg)** defines the angle of the excitation.
  - TONESPEC defines the symbolic frequency of the excitation in HB analysis.
    - f1 is tone-1, f2 is tone-2 , …, fn is tone-n
    - Port can also excite the circuit at mixing products, for example, f1-f2
- Bias can be defined as in Bias-type ports.

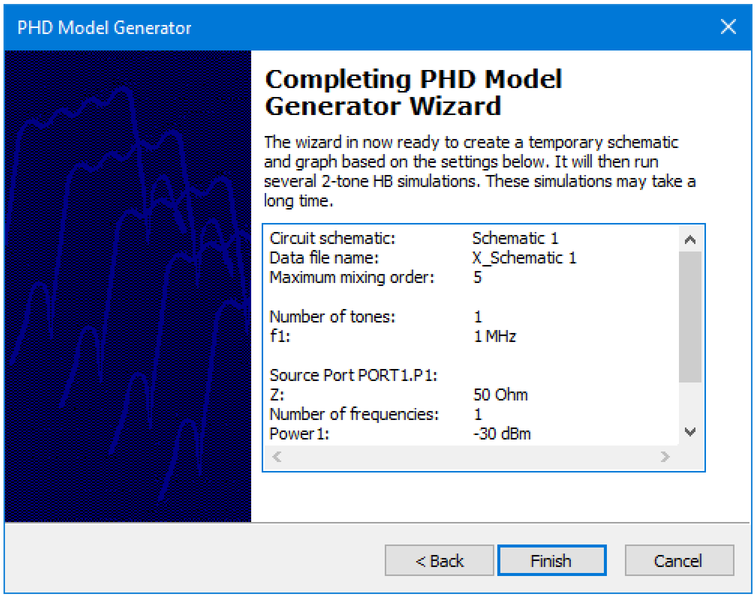
If the port type is **Load**:

- Z (ohm) specifies the port impedance.
- Load type can be either impedance or gamma in real/imaginary or magnitude/angle formats.
- NFreq defines the port impedance at different frequencies.
- LoadN defines the port impedance at frequency N. There are NFreq Load definitions in total (Load1, Load2, …, LoadN).
  - Depending on the load type and format, both the impedance (or gamma), real part (or magnitude), and imaginary part (or angle) can be swept.
  - TONESPEC defines the symbolic frequency of the load.
- Bias can be defined as in Bias-type ports.

If the port type is **Bias**:

- Bias defines the port bias feed. Type can be **None** (no bias at port), **Bias voltage** or **Bias current**.
- If **Sweep** is selected, the bias voltage or current value is defined by **Min**, **Max**, and **Steps**.

4. Click **Finish** to create a new circuit schematic with an XPARAM block that uses the extracted PHD model data.

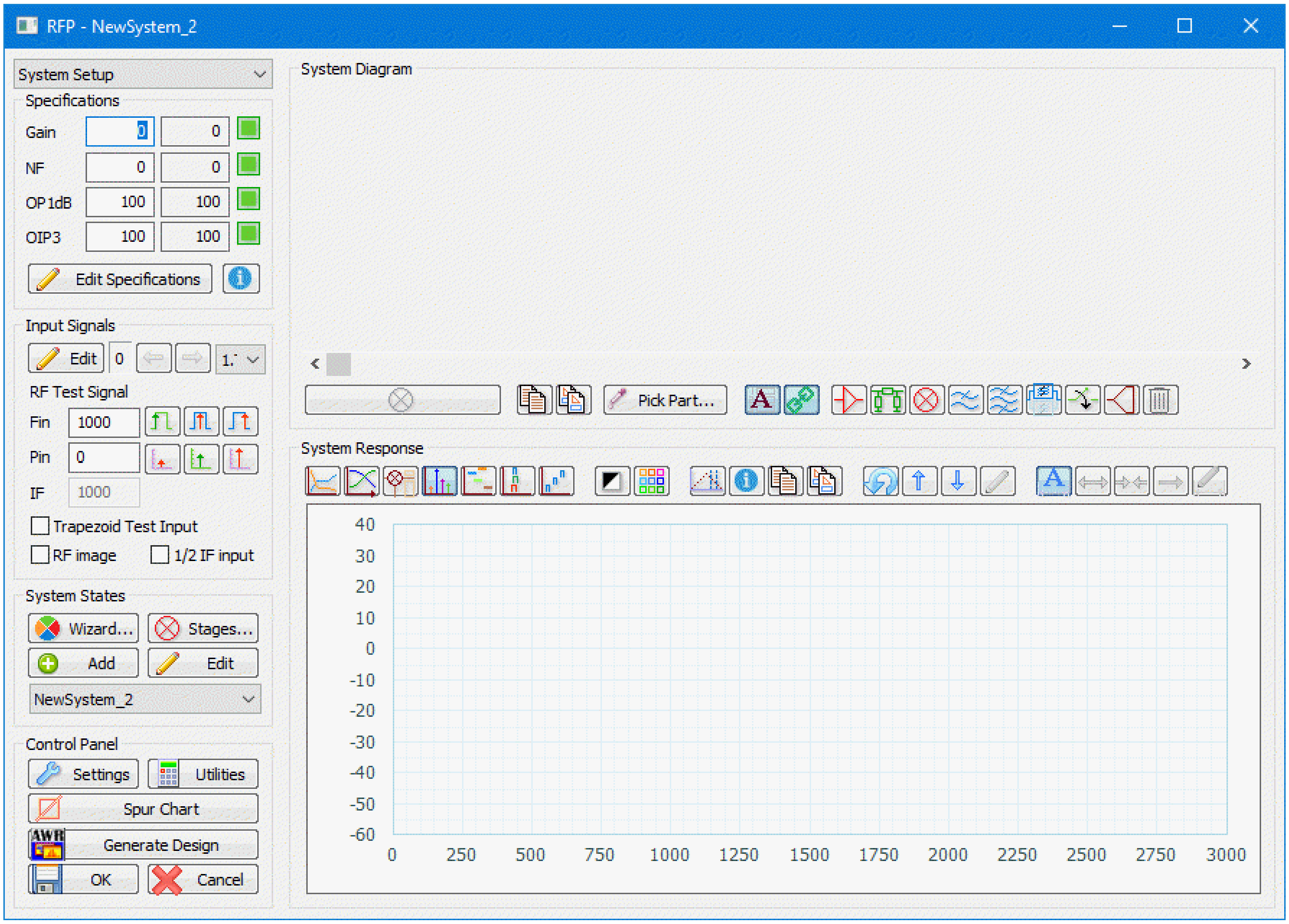


13.15. RFP RF Planning Tool Wizard

The Cadence RF Planner™ software module, a frequency planning synthesis tool used via the RFP RF Planning Tool Wizard, allows you to determine spurious free bandwidths. This wizard is an essential analysis tool when developing radio communications systems. It surpasses common spreadsheet analysis calculations and displays clear results in several formats. In addition to showing the power levels and frequencies of the signals, the root causes of the signals also displays. In addition to spur analysis, the RFP gives you the first cut of cascaded measurements such as NF, P1dB, SNR, and IM3, as well as spurious free dynamic range. The RFP is also seamlessly integrated with VSS software. You can generate designs in the AWR Design Environment platform as VSS projects for further detailed analysis and optimization.

Using the RFP and VSS software to determine system specifications is better than a traditional spreadsheet-only method. VSS software provides a richer set of models, and calculations account for real world effects. In addition, yield analysis and optimization are available, and in-depth spur analysis is built in.

To access the RFP, open the **Wizards** node in the Project Browser and double-click **RFP RF Planning Tool**. The main RFP dialog box displays as shown in the following figure.



This first section of this chapter provides general information about the RFP Radio Frequency Planning Wizard through definitions, user interface graphics, and option descriptions. The second section includes an example that demonstrates the RFP tool's capabilities.

13.15.1. RFP RF Planning Tool Basics

Every RFP subnode of the **RFP RF Planning Tool** node in the Project Browser is an instance of the RFP wizard that contains one design object. A design object contains one or more system objects (also called "system states"). Most of the time, the main RFP dialog box displays the contents of the selected system, although there are other means and dialog boxes to access and modify all systems in the design. Each system object contains:

- System Budget Specifications, which define the target values for gain, noise figure, compression point and other similar parameters for RF budget calculation. Budget specifications are located at the top left of the main RFP dialog box.
- System Diagram, where components are cascaded. There is no restriction or certain template requirement for the order of components. They can be added in any order and RFP does necessary budget calculations, as well as finds a suitable frequency conversion scheme for a given order. The system diagram displays at the top of the main RFP dialog box. Each component is represented with a button you can click to edit. Some major parameters are also listed beneath the component buttons.

- Input Signal Bands, (or Input Frequency Bands, or Input Bands), which define the RF inputs to each system. An input band is defined by three main values: min/max frequency and power level. There may be more than one input band for a system, and they can also be designated as threats. RFP has the capability to auto-adjust various parameters of components. One of the input bands is therefore called a "selected band", whose frequencies the wizard uses to auto-adjust LO's and IF frequencies.
- Conversion Scheme, which defines how the RF to IF conversion through one or more mixer stages is performed. For example, in a two-stage frequency conversion system, one may take either RF-LO or RF+LO for the first IF. Depending on the conversion scheme chosen, RFP tries to auto-adjust all relevant LO's and IF's.

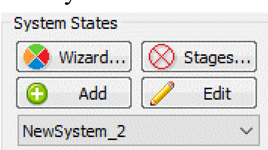
The RFP RF Planning Tool Wizard has the following system component types:

- AMP - Amplifier
- ATT - Passive or active attenuator
- LPF - Lowpass filter
- BPF - Bandpass filter
- MIX - Mixer
- SWT - RF Switch
- SBP - Switched Bandpass Filter
- ADC - Analog to Digital Converter

For details on parameters of components, see the Edit dialog boxes for the components.

13.15.2. Maintaining System States

The System States section of the main RFP dialog box provides access to all system states for editing.

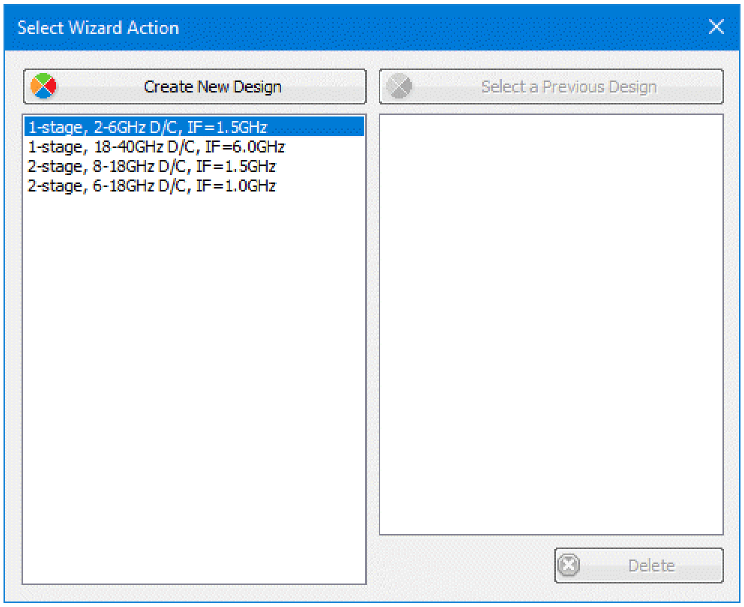


The **Wizard** button displays the "Select Wizard Action Dialog Box".

The **Stages** button displays the "System States - Conversion Stages Dialog Box" where you can edit input bands, and LO and IF frequencies of all systems in one place. For channelized downconverters, input bands are assigned to each channel. You can optimize input bands through this dialog box, where bands are easily edited and re-divided into channels.

13.15.2.1. Select Wizard Action Dialog Box

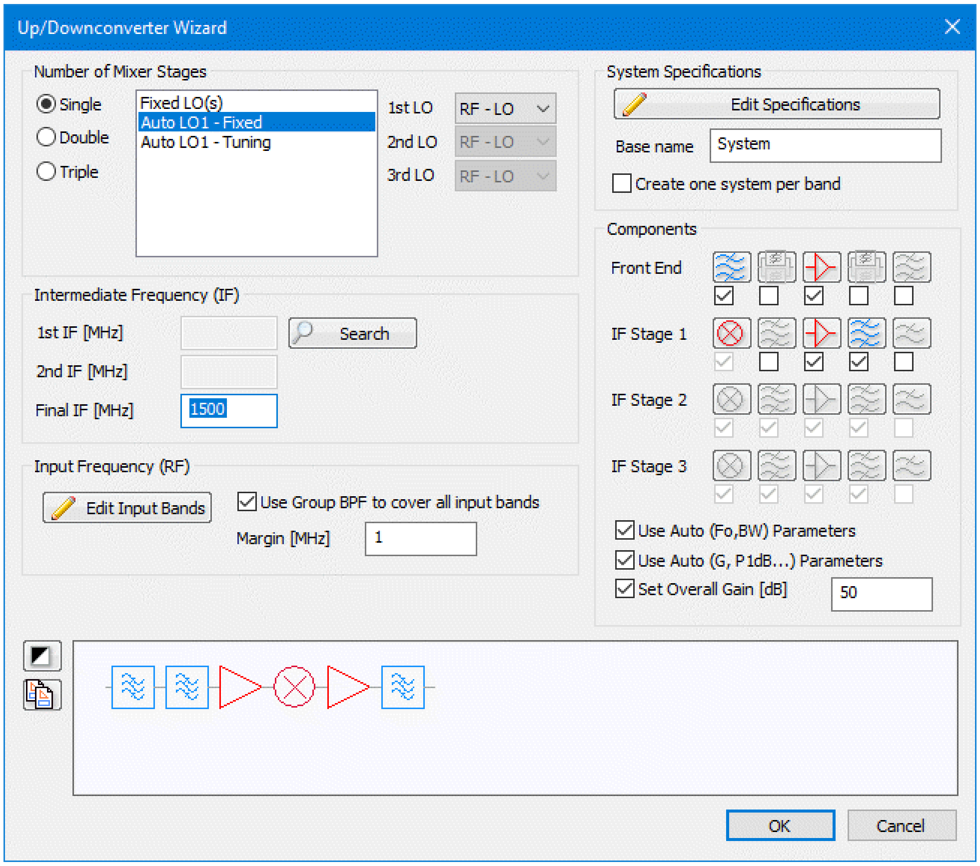
The Select Wizard Action dialog box includes buttons for creating pre-stored channelized frequency up/downconverter examples, as well as previously created designs. To access this dialog box, click the **Wizard** button in the main RFP dialog box.



- **Create New Design** - Loads the selected built-in conversion settings into the wizard. These are typical frequency conversion examples that are pre-stored in the program.
- **Select a Previous Design** - Loads the existing selected custom design. Each time the wizard runs, the design is stored and listed in the right pane of the dialog box for easy access.
- **Delete** - Deletes the selected custom design from the pane on the right.

13.15.2.2. Up/Downconverter Wizard Dialog Box

The Up/Downconverter Wizard sets up RFP systems using the specified configuration settings. When you start this wizard, the diagrams and input bands in the main design dialog box are replaced with the new content as specified here. To access this dialog box, click the **Create New Design** button in the "Select Wizard Action Dialog Box".



Under **Number of Mixer Stages**, select the number of conversion stages and the type of IF output as **RF-LO**, **RF+LO** or **LO-RF**. The LO behavior is selected from one of the following: **Fixed LO(s)** where LO's are user-specified and kept fixed, so the RF band maps to an IF band with the same bandwidth. **Auto LO(x) - Fixed** where the selected auto LO is automatically set, so the center of the RF band maps to the Final IF frequency and is fixed. The RF band maps to an IF band with the same bandwidth. **Auto LO(x) - Tuning** where LO is calculated to map every single frequency in the RF band into the single Final IF frequency. The RF band maps to a spot IF frequency.

Under **Intermediate Frequency (IF)**, select IF centers as appropriate. Since only one of the LO(s) and IF(s) can be automatic, you must specify all other frequencies . RFP enables the frequencies that you should specify. Click the **Search** button to display the "LO/IF Search Dialog Box".

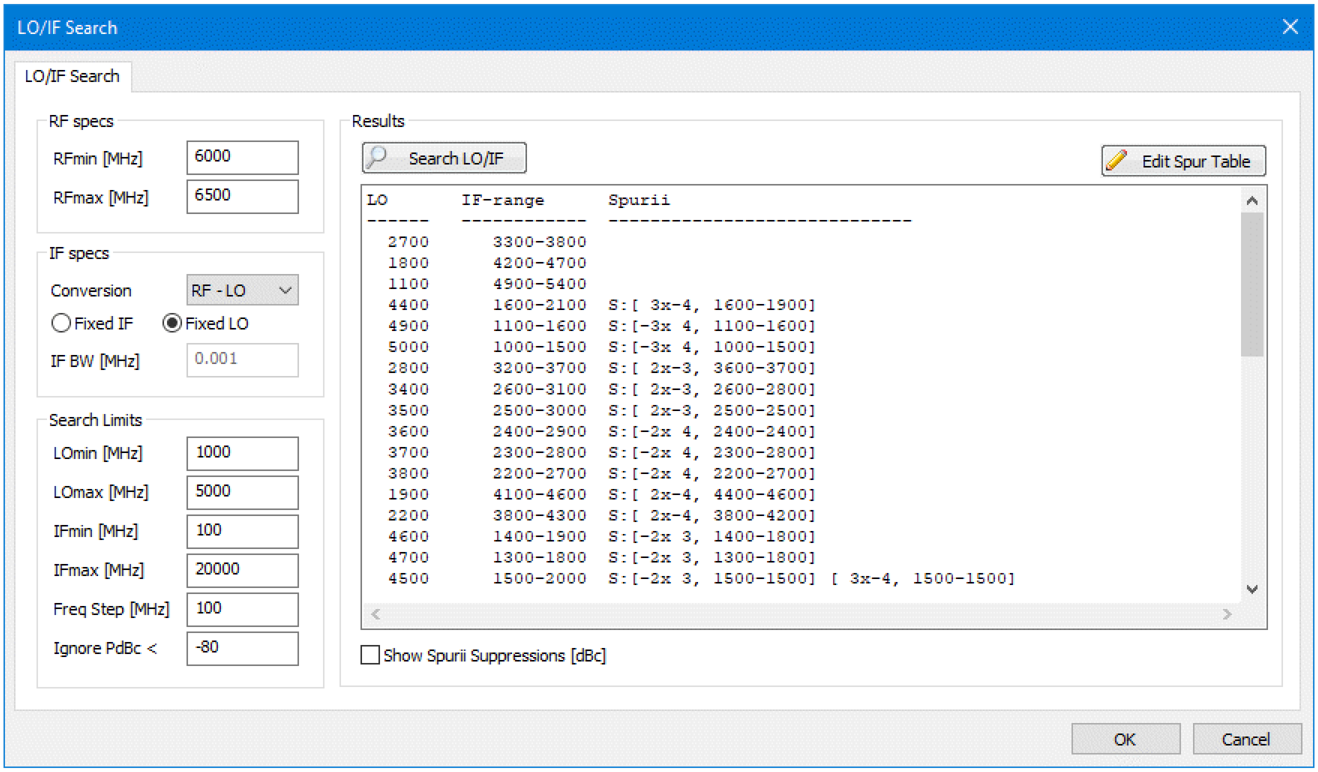
Under **Input Frequency (RF)**, specify the input RF bands that can be threatening or friendly. Select the **Use Group BPF to cover all input bands** check box to add an RF filter at the front end. Specify in **Margin** how much the filter must be extended on each side of the overall RF input range.

Under **System Specifications**, you enter specifications by clicking the **Edit Specifications** button. The **Base name** labels the system(s) with enumeration. If the **Create one system per band** check box is selected, RFP creates one system for each RF input band and assigns that band to the system as the active, friendly input. If this check box is cleared, only one system is created.

Under **Components**, you can select and edit the components you want to use. If **Use Auto (Fo,BW) Parameters** is selected, all filter frequency settings are turned into auto, and they are set to facilitate the intended RF/IF conversion. If this check box is cleared, the parameters remain as entered. Select the **Use Auto (G,P1dB...) Parameters** check box to automatically set all components with these parameters to auto mode. Select the **Overall Gain[dB]** check box to set the RF/IF conversion gain of the system.

13.15.2.3. LO/IF Search Dialog Box

The LO/IF Search dialog box is used for searching optimum, spur-free LO or IF frequencies. To access this dialog box, click the **Search** button in the "Up/Downconverter Wizard Dialog Box".



To use this dialog box you first enter an RF input frequency range under **RF specs**, then select one of the following IF **Conversion** schemes:

- **RF – LO**
- **RF + LO**
- **LO – RF**

Select either **Fixed IF** or **Fixed LO** as follows:

- For **Fixed IF**, LO is swept according to RF to keep IF constant. **IF BW** determines the output window where the spurii is searched within. In practice, a bandpass filter follows the frequency conversion element (mixer) with a certain bandwidth, IF BW, which is only wide enough to allow data.
- For **Fixed LO**, IF is calculated per the selected **Conversion** scheme. As a result, IF varies within a bandwidth that is the same as the RF bandwidth. In practice, the bandpass filter that follows the mixer has a fixed center frequency and bandwidth. The center frequency is calculated for the RF center frequency using the **Conversion** scheme. **IF BW** , where the spurii is searched within, is also equal to the RF bandwidth. As an example: For RF=3000-4000MHz and IF=RF-LO, when LO=2500MHz, IF varies between 3000-2500=500MHz and 4000-2500=1500MHz. This results in a fixed IF window of (500-1500MHz), which is centered at 1000MHz with 1000MHz bandwidth.

Under **Search Limits**, the options are used to limit the LO and IF ranges for search:

- **LOmin**, **LOmax**, **IFmin** and **IFmax** determine the range of values LO and IF can take. They constraint the values when used as either a sweep variable or a resultant variable.
- **Freq Step** is used as the increment value for search.
- **Ignore PdBc** gives the threshold below which it is ignored as noise.

Click the **Edit Spur Table** button to display the "Spur Table Dialog Box" and set the spur level criteria, and then click the **Search LO/IF** button to search.

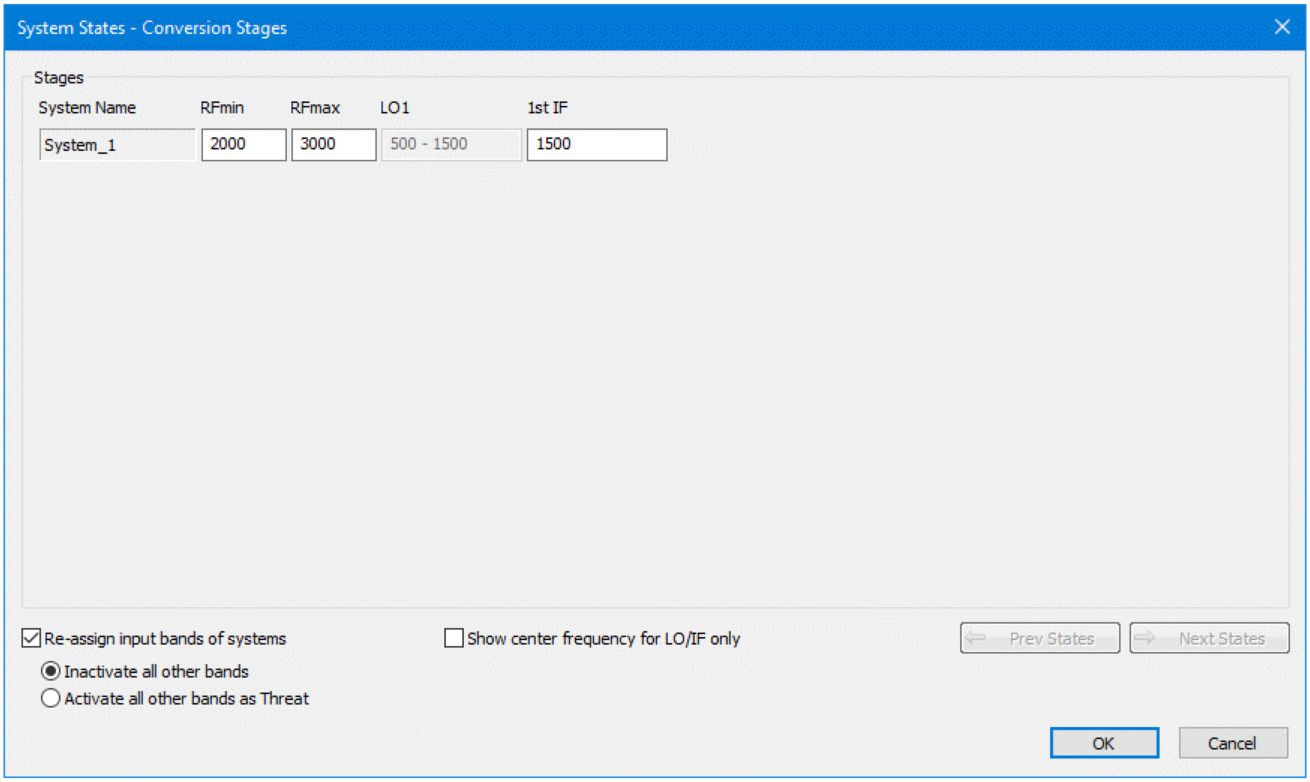
In an example with **RFmin** set to 6000, **RFmax** set to 6500, **Conversion** set to **RF-LO**, **Fixed LO** case, **LOmin** set to 1000, **LOmin** set to 5000, **IFmin** set to 100, **IFmax** set to 20000, **Freq Step** set to 100, and **Ignore PdBc** set to -80, since LO is fixed, IF is swept. The IF range to sweep is 100 to 20000MHz with a step of 100MHz. For each iteration of IF, LO is calculated from IF=RF-LO (or LO=RF-IF) and checked if it is between 1000-5000MHz. If so, it is recorded as a valid LO frequency and the resultant mixer spurs are calculated if they fall within the IF window. The IF window is centered at IF iteration value and its bandwidth is 500MHz (RF bandwidth). If the IF window cannot fit to positive frequencies, it is not a valid result, so the iteration is skipped to the next IF value. For example, when IF = 100MHz, a bandwidth of 500MHz cannot fit. In the previous example, the first valid IF frequency is 300MHz, where the IF window lies within (300-500/2=150) to (300+500/2=550). The first few lines of results for the search are:

```
LO                  IF-range                    Spurii
------              ------------                ------------------------------
2700,               3300-3800
1800,               4200-4700
1100,               4900-5400
4400,               1600-2100 S: [ 3x-4,         400-1900, -77]
4900,               1100-1600 S: [-3x 4,         100-1600, -77]
```

The results are given as sorted by the most spur-free frequencies. In the previous example, when LO is 2700MHz, 1800MHz, or 1100MHz, the corresponding IF windows do not contain any spurii above -80dBc. With slightly worse LO selections: LO=4400MHz, 4900MHz results in a 3x4 component at a level of -77dBc. The LO/IF Search utility can quickly search for a wide range of frequencies for spur-free conversions, and it is a strong alternative to the popular Spur Chart method.

13.15.2.4. System States - Conversion Stages Dialog Box

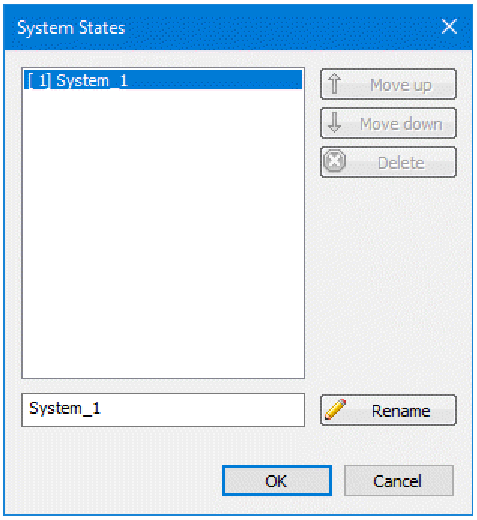
The System States - Conversion Stages dialog box provides access to the RF inputs, LO's, and IF's of all systems in one location. To access this dialog box, click the **Stages** button on the main RFP dialog box. This dialog box is mainly used for, but not limited to, channelized converter designs. Systems are listed together with their current active input bands, LO and IF set frequencies. If a system has a fixed LO, that LO is available to edit, otherwise, the corresponding IF is available for editing. For channelized converters, input bands are covered by channels in a contiguous manner. In RFP, channels are represented by systems. In this dialog box, you can change input bands for each system. For example, you can widen the System 1 band and narrow the System 2 band while keeping the bands contiguous.



Select the **Re-assign input bands of systems** check box if the RF bands edited in the dialog box are intended to re-assign all systems. It is best to give an example for how the bands are reassigned. For example, the input band (RF band) for System 1 is called Band 1, the input band for System 2 is called Band 2, and so on, so there are N bands for N systems. When you select this option and **Inactivate all other bands as Threat**, System 1 has only one active band (Band 1) and all the other bands are inactive for this system. Similarly, System 2 has one active band (Band 2) with all the other bands set to inactive. When you select **Re-assign input bands of systems**, and **Active all other bands as Threat**, System 1 has one active band (Band 1) and N-1 threat bands (Band 2 to Band N). Similarly, System 2 has one active band (Band 2) and N-1 threat bands (Band 1 and Band3 to Band N). When you select **Show center frequency for LO/IF only**, the LO/IF frequency ranges only display as center frequencies, which is easier to interpret in some cases. This dialog box display only ten systems at a time. To see additional systems, click the **Prev States** or **Next States** buttons.

13.15.2.5. System States Dialog Box

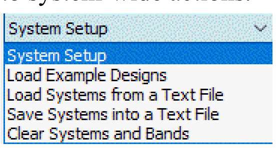
The System States dialog box is used to sort, rename or delete systems from a design. To access this dialog box, click the **Edit** button under **System States** in the main RFP dialog box.



Click the **Move up** or **Move down** buttons to change the index of the selected system in the design. The systems are listed in the dialog box by their index. System indices are used for re-assignment of input bands, plotting responses, and reporting plot information. Their names are display only. For example, if you select System_3 and move it up in the list, it is named and treated as [System#2] internally although the name is still System_3. Similarly, in this case, System_2 is moved down to third place, and is treated as [System#3] internally. To delete the selected system, click the **Delete** button. To rename the selected system, enter the name in the edit box, and click the **Rename** button.

13.15.2.6. System Setup Shortcuts

A **System Setup** drop-down menu is available at the top left of the main RFP dialog box to provide you with shortcuts to system-wide actions.



Select from the following options:

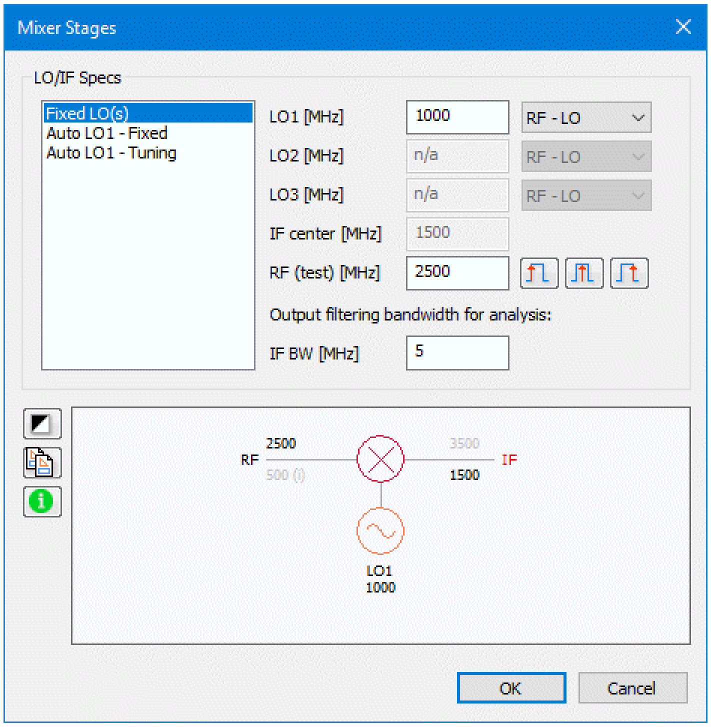
- **Load Example Designs** to display the Select Wizard Action dialog box.
- **Load Systems from a Text File** to load system setup from a text-based file. The file can be externally edited in Notepad if needed.
- **Save Systems into a Text File** to save system setup to a text-based file. The file can be externally edited in Notepad if needed. When problems occur, this is a convenient way to check the wizard data and exchange with Cadence Support if needed.
- **Clear Systems and Bands** to reset the working RFP environment by clearing all user-edited settings.

13.15.3. Maintaining the Selected System

The following sections include information on dialog boxes that provide system control and configuration.

13.15.3.1. Mixer Stages Dialog Box

The Mixer Stages dialog box is used to change the frequency conversion scheme of the current system. To access this dialog box, click the **Mixers** button under **System Diagram** in the main RFP dialog box.



Frequency conversion to an intermediate frequency (IF) occurs by adding RF and LO or by subtracting one from the other. For a single conversion RF-LO case, the equation is: IF = RF - LO. RF is already specified by input bands. IF and LO are therefore interdependent.

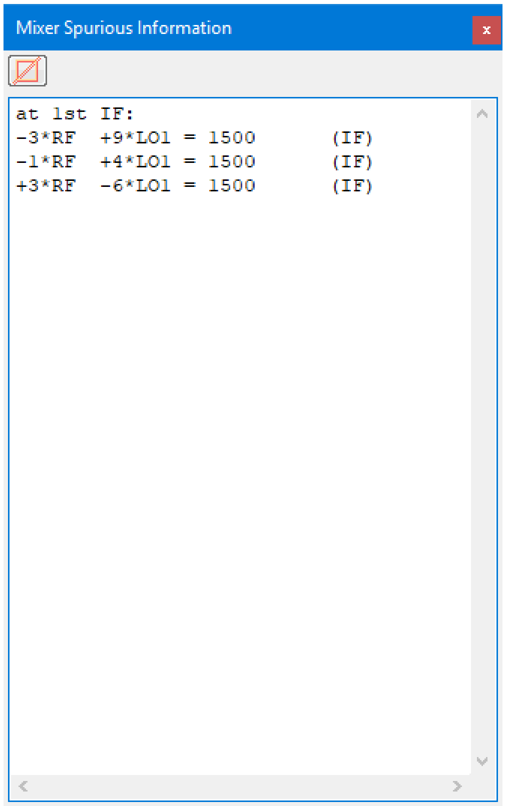
The way RFP functions depends on your option selection:

- **Fixed LO(s)** - where LO's are manually entered and kept fixed. RF band maps to an IF band with the same bandwidth.
- **Auto LO(x) - Fixed** - where the selected auto LO is automatically set so that the center of RF band maps to Final IF frequency and kept fixed. RF band maps to an IF band with the same bandwidth.
- **Auto LO(x) - Tuning** - where LO is calculated to map every single frequency in RF band into the single Final IF frequency. RF band maps to a spot IF frequency.

For double and triple conversion, similarly, there is always one parameter that must be free from specification and it is calculated by using other variables. They are called Auto LO1, Auto LO2, Auto LO3, or Fixed LO(s). When Auto LO2 is selected, all other variables (for example LO1 and IF) are available for editing, and LO2 is automatically calculated from the edited variables. In the simplified system diagram at the bottom of the dialog box, the free variable is displayed in red. A test case is provided for easy interpretation of the conversion scheme. RF test frequency, which is also available in the main RFP dialog box, is input to the system and the basic conversion frequencies are displayed in the simplified system diagram. As in any mixer conversion, there are two major IF outputs: difference and addition of RF and LO. In the dialog box, they are shown after each mixer in two different colors. The IF that belongs to the selected scheme for the mixer is shown in black/white, and the other is shown in gray. To select the grayed IF output, select the corresponding conversion from the drop-down box provided in the same row as that LO, or simply click the grayed output. If an **IF BW** entry is used for checking the spurii for the selected conversion scheme, it determines the bandwidth of the IF window, where the spurii is checked to fall into. You can view Mixer spurious information by clicking the **i** (Information) button to toggle its display in a separate "Mixer Spurious Information Window".

13.15.3.2. Mixer Spurious Information Window

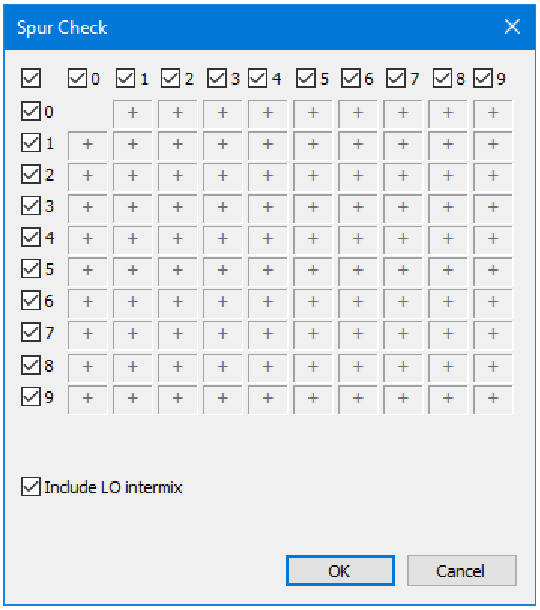
The Mixer Spurious Information window displays the problematic frequencies of the selected mixer conversion scheme. To access this dialog box, click the**i** (Information) button in the "Mixer Stages Dialog Box".



When RF, LO, and/or IF of the conversion scheme are given, an IF window is determined around the desired/given IF. The IF window bandwidth (**IF BW**) is specified in the Mixer Stages dialog box. A spur search is then performed and any spurious component that falls into the IF window is reported. The location in which the spur occurs is also reported as 1st IF, 2nd IF, and so on. You can toggle the order of spur table entries for calculation in or out by clicking the **Edit Spur Check Options** button at the top of the window to display the "Spur Check Dialog Box".

13.15.3.3. Spur Check Dialog Box

The Spur Check dialog box toggles the mixer spur components in or out of the spur search. To access this dialog box, click the **Edit Spur Check Options** button at the top of the "Mixer Spurious Information Window".



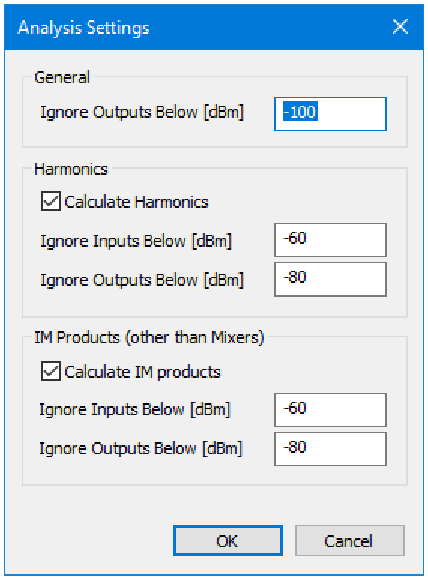
By clicking the check boxes next to rows or columns, the entire m or n of the mRF+nLO is included or excluded in the search. When a cell displays a "+", that (m,n) cell is used in the spur search. The rows are for m (RF) and the columns are for n (LO).

If you select the **Include LO intermix** check box, combinations of LO's are included in the spur check. In practical converters, LO's can leak through stages to mix with other LO's to create false IF. For example, in a double conversion design, a 20dBm 1st LO leaks through the 1st mixer by 30dB attenuation. It is further attenuated by 60dB in the bandpass filter after the 1st mixer. This means that the 1st LO presents itself as an RF input to the 2nd mixer as 20-(30+60) = -70dBm. The mixer can then generate a false IF output by mixing -70dBm LO1 and LO2, even when RF is terminated by a load. Select this check box to observe in the Mixer Spurious Information Window if there is any possibility of LO intermixing issues.

The **Include (2nd, 3rd) LO mixing with RF** check box is used to include leakage of 2nd and 3rd LO's into 1st stage in spur checks. When selected, the spur search algorithm treats 2nd LO and 3rd LO as 1st LO and calculates spurii output of them and the RF input. If they fall into the 1st IF, they are reported as problematic. This situation occurs in systems where 2nd and 3rd LO are not well isolated from the 1st mixer's LO port. They behave like LO and their mixing with the RF output may create in-band IF products that are not possible to remove.

13.15.3.4. Analysis Setting Dialog Box

The Analysis Settings dialog box is used to change global settings for spectrum calculation. To access this dialog box, click the **Settings** button in the main RFP dialog box.



This dialog box mainly gives limits below which signals are ignored from analysis. If the limit is for an input, then any signal level below the given limit is ignored as an input. If the limit is for an output, then any signal under that power level is omitted from reports or from inputs to the next component.

Under **General**, **Ignore Outputs Below** sets the minimum level of a signal at any output to be counted as a "high-enough" signal for the next stage.

Select the **Calculate Harmonics** check box to include harmonics of frequency outputs to be used in analysis. 2nd- and 3rd-order harmonic outputs are calculated by using OIP2 and OIP3:

```
2nd harmonic output = 2*Pout - OIP2 - 6
3rd harmonic output = 3*Pout - OIP3 - 9.54
```

These are only linear approximations of harmonics. When a component is in output compression or in saturation for the fundamental frequency, harmonics are extremely difficult to calculate and there is no generic way, so you should use harmonic outputs for guidance only.

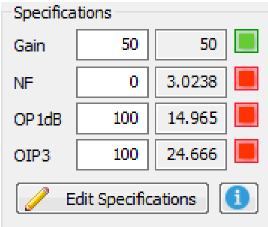
Harmonics are calculated for all components except LPF, BPF and MIX.

Click the **Calculate IM products** check box to include inter-modulation calculation in the analysis for all active components. IM products for mixers are always calculated. This option is mainly used for amplifiers with high-input signals. The wizard calculates IM products of high-power inputs in combinations of 2 at six frequencies:

```
F1-F2, F2-F1, 2*F1-F2, 2*F2-F1, 2*F1+F2, 2*F2+F1
```

13.15.3.5. Specifications Group

The Specifications group presents major system budget specifications and a button to access the System Specifications dialog box to edit them in full.



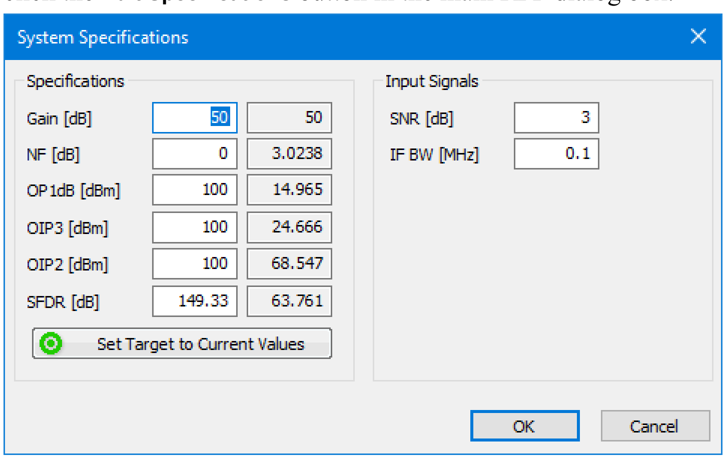
This group has two columns: the values on the left show the target specifications, while the values on the right show the actual values calculated for the system. An icon next to each row changes colors as targets are reached, approached, or missed. When a target value is reached or exceeded, the icon displays in green. If the actual value is slightly less than the target, the icon displays in orange. When the actual value reasonably falls short of the target, the icon displays in red.

You can directly edit values in this group. Due to the limited display area, not all of the budget specification parameters are displayed. To edit all of the specs, click the **Edit Specifications** button to display the System Specifications dialog box.

If attenuation or gain values of components are in **Auto** mode, the RFP RF Planning Tool Wizard tries to adjust them to match the overall gain specification by distributing the gain equally between auto-stages. For example, if G=10dB is specified and the system diagram is set up using a mixer with CL=6 and two auto-gain amplifiers, the gain of the amplifiers is set to 8dB each to make 16dB gain in cascade, which reduces to the desired 10dB when combined with the mixer's conversion loss.

13.15.3.6. System Specifications Dialog Box

The System Specifications dialog box allows you to edit system budget parameters in detail. To access this dialog box, click the **Edit Specifications** button in the main RFP dialog box.



The two columns in the **Specifications** group display target and current values. You can edit the target values. The current values are calculated from the current system and are display only. Click the **Set Target to Current Values** button to set the target values to the current values. This auto-set is useful, especially when you want to set the system state as a reference and view changes in budget parameters when the system changes.

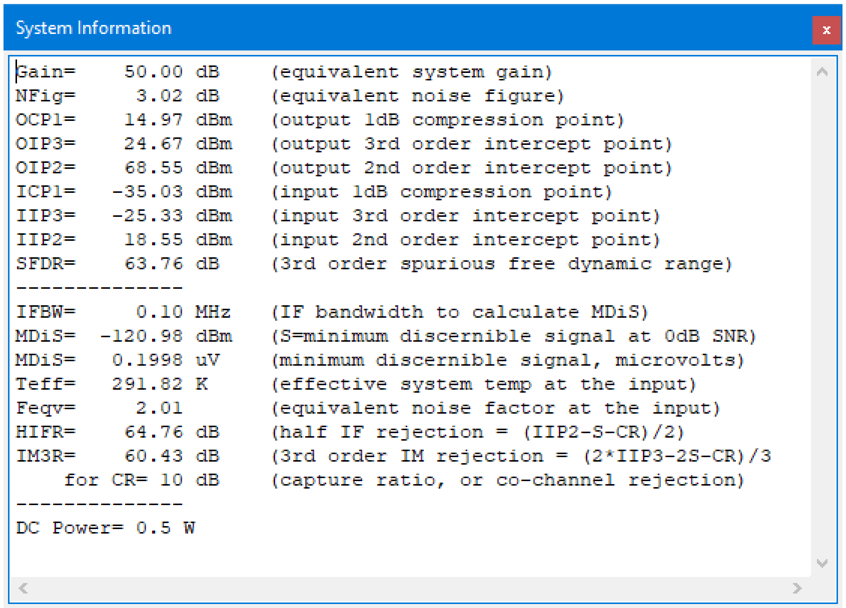
SNR and **IF BW** are used to calculate minimum discernible signal (MDS) and dynamic range. MDS and SFDR are calculated using the following equations:

```
MDS = -174 + 10*log10 (IFBW) + NF + SNR
SFDR = Phigh - Plow = (MDS + 2*IIP3)/3 - MDS = 2/3 * (IIP3 - MDS)
```

where **IF BW** is in Hz.

13.15.3.7. System Information Window

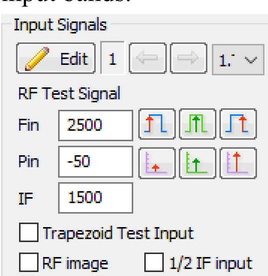
The System Information window displays the results of system budget calculations. To open or close this window, click the **i** (Information) button in the **Specifications** group in the main RFP dialog box to toggle its display.



This window shows live information with term descriptions. For more information, see *RF Design Guide*, a book by P. Vizmuller.

13.15.4. Maintaining Input Bands

Every system has input frequency bands. The **Input Signals** group in the main RFP dialog box contains options to maintain input bands.



To edit frequency band details, click the **Edit** button to display the Input Signal Bands dialog box. Additional options in the **Input Signals** group provide shortcuts to input band properties.

The text box next to the **Edit** button shows the index of the selected input band.

To select the previous or next input band, click the **Previous Signal Band** or **Next Signal Band** arrow buttons to the right of the text box.

The option to the right of the arrow buttons includes three options for band selection:

- **Traverse active bands only** selects among the bands designated as active.
- **Traverse all bands, making unselected bands Inactive** selects among all of the input bands, each time setting only one input band active and all others inactive.
- **Traverse all bands, making unselected bands Threat** selects among all of the input bands, each time setting only one input band active and all others active but threat.

Every input band contains frequency and power ranges. Input bands also have a test frequency and test power that vary in the defined ranges. Power level is used in band or spot frequency analysis, whereas test frequency is only used in spot frequency analysis. The test frequency and power display as **Fin** and **Pin**. You can change these by typing new values, or by clicking in the option and then scrolling with the mouse wheel to change values. The buttons next to **Fin** and **Pin** are shortcuts for setting test frequency or test power to minimum, center, or maximum values in the range. The button color changes to green when you apply its test values.

IF displays the final IF frequency calculated for the test signal. When the conversion is any of the Auto LO(x) schemes (for example, fixed IF), you can edit the frequency here instead of opening the Mixers Stages dialog box.

Selecting the **RF Image** check box allows you to include RF image input for the first mixer in the system as a threat input. RF image for conversion schemes are:

```
For IF = RF - LO, RF image input = LO - IF
For IF = RF + LO, RF image input = LO + IF
For IF = LO - RF, RF image input = LO + IF
```

Example: IF=RF-LO scheme is selected. RF=5000-6000MHz, LO=4000MHz, and RF test (Fin)=5500MHz.

In the spot frequency analysis, the intended IF=RF-LO=5500-4000=1500MHz. So, RF image occurs at LO-IF=4000-1500=2500MHz. If RF image is input to the system as a threat, it generates the same output as the intended IF: IF=|RFimage-LO|=|2500-4000|=1500MHz.

In the frequency band analysis, RF image is included as a threat band. RF image frequency for 5000MHz input is 3000MHz. Similarly, RF image frequency for 6000MHz input is 2000MHz. As a result, the RF image as a threat band is 2000-3000MHz.

When input bands pass mixer conversion stages in the LO-RF mode, they are inverted at the IF output. When a few mixing stages are combined, it becomes difficult to tell the inverted and non-inverted spectrum. You can select the **Trapezoid Test Input** check box to put a slant on the left-hand side of the RF input band so that the processed spectrum is more easily distinguished.

When you select **RF Image**, the threat input is automatically calculated and input in either spot frequency or frequency band mode.

Selecting the **1/2 IF input** check box allows you to include RF threat input that generates half IF frequency at the first mixer output. Since the mixer generates multiples of differences as well, a 2nd order product of half IF falls exactly onto IF. Half IF threat inputs for conversion schemes are as follows:

```
For IF = RF - LO, Half IF input = LO + IF/2
For IF = RF + LO, not practical
For IF = LO - RF, Half IF input = LO - IF/2
```

Since this threat input is only IF/2 away from LO, it is difficult to remove half IF input by RF filtering. In practice, RF and LO are kept as far as possible to build immunity against half IF input.

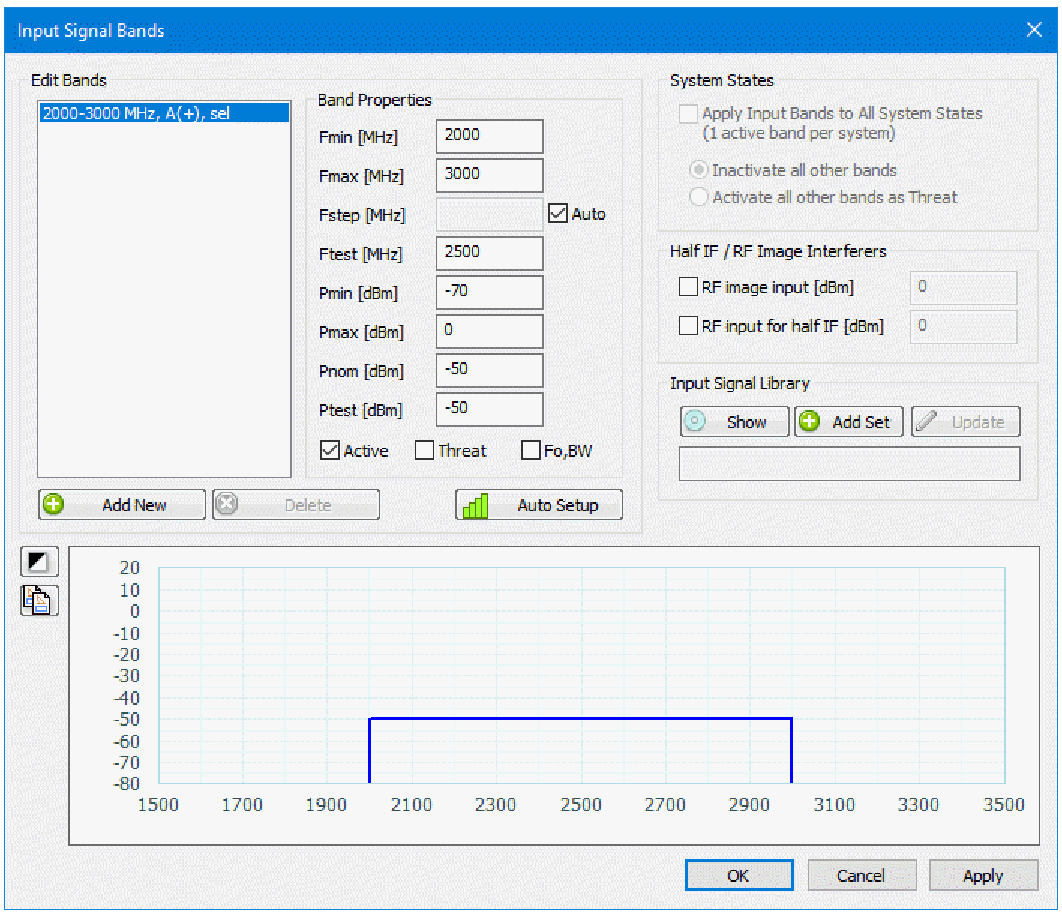
Example: IF=RF-LO scheme is chosen. RF=5500-6000MHz, LO=4500MHz, and RF test (Fin)=5500MHz.

In the spot frequency analysis, the intended IF=RF-LO=5500-4500=1000MHz. So, half IF input occurs at LO+IF/2=4500+1000/2=5000MHz. If half IF threat is input to the system, the mixer's 2*(RF-LO) generates 2*(5000-4500) = 1000MHz, which is exactly the intended IF.

In the frequency band analysis, half IF is included as a threat band. Half IF frequency for 5500MHz input is 4500MHz. Similarly, half IF frequency for 6000MHz input is 5250MHz. As a result, half IF input as a threat band is 4500-5250MHz. This example system normally has a pre-selection filter for 5500-6000MHz. With the mixer suppression, half IF input should be suppressed below system sensitivity so that it does not affect the dynamic range. A typical mixer suppression for 2*(RF-LO) product is 60dBc. If the maximum threat input power is 20dBm, and the sensitivity is -90dBm, the filter suppression is found as: 20dBm-60dB-S < -90dBm, which yields S > 50dB. 50dB suppression at just 250MHz away from the passband corner is a challenge. To overcome this, you need to choose a mixer with a high 2x2 suppression or a different LO.

13.15.4.1. Input Signal Bands Dialog Box

The Input Signal Bands dialog box is used to specify input signal frequency bands to a system. To access this dialog box, click the **Edit Bands** button in the **Input Signals** group in the main RFP dialog box.



Bands are listed in the **Edit Bands** group. Each band has frequency and power ranges, which have minimum, maximum, center, test and step values.

In the **Band Properties** group, **Fmin** and **Fmax** are the lower and upper frequency of an input band. In the frequency band analysis, these values determine the width of the input signal. In the spot frequency mode, **Ftest** (**Fin** in the main RFP dialog box) represents input signal frequency. **Fstep** is the step frequency used as the increment when clicking in the **Fin** option and then scrolling with the mouse wheel to change values. When you select the **Auto** check box, the program determines the frequency step automatically. Nominal frequency, Fnom, is not specified in this dialog box; it is automatically determined to be the center of the frequency range.

Pmin and **Pmax** are the lower and upper bounds of the power level. They do not contribute to analysis. In the main RFP dialog box, when you change **Pin** with the mouse wheel, these extrema come into effect to limit the value of **Pin**. **Ptest** is the test power of the input signal in both spot frequency and frequency band analysis. **Pnom** is the nominal power. When you click the **Set Test Power to Nominal Power** button next to **Pin** in the main RFP dialog box, **Ptest** is set to **Pnom**. Pstep is 1dB by default, but it is not specified in this dialog box.

Each band can be active or inactive (disabled). Active signals are always input to system during analysis. However, only one band is selected for the system to auto-adjust frequencies. To specify that band, select it and then click **OK**. Alternatively, click the **Prev Signal Band** or **Next Signal Band** button in the **Input Signals** group in the main RFP dialog box.

To activate a band for analysis, select the **Active** check box. Select the **Threat** check box to designate the selected band as a threat band. A band can be active but not threat; threat but not active; or inactive, so select check boxes accordingly.

By default, **Fmin** and **Fmax** are specified for frequency range. If you want to specify center frequency and bandwidth instead of corner frequencies, select the **Fo,BW** check box.

To add a new input band, click the **Add New** button.

To delete the selected input band, click the **Delete** button.

To set up multiple input bands with the same bandwidths and separated by the same frequency gap, click the **Auto Setup** button to display the Input Bands Auto Setup dialog box.

The **Half IF/RF Image Interferers** group contains options to generate half IF and RF image input threats to systems. The check boxes in this group activate the relevant threat band, while the edit boxes specify their power level. For details on interferers, see “Maintaining Input Bands”.

The Input Signal Bands dialog box applies to one system, however the **System States** group includes options to apply the input bands to all systems in the design. When you select the **Apply Input Bands to All System States** check box, Band 1 is assigned to System 1 as the selected band, Band 2 is assigned to System 2, Band 3 to System 3 and so on. Remaining bands are assigned either as threat or inactive. If you select **Inactivate all other bands**, then Bands 2 to N are set as inactive for System 1; Band 1 and Bands 3 to N are set as inactive for System 2, and so on. If you select **Activate all other bands as Threat**, those bands are activated and set as threat.

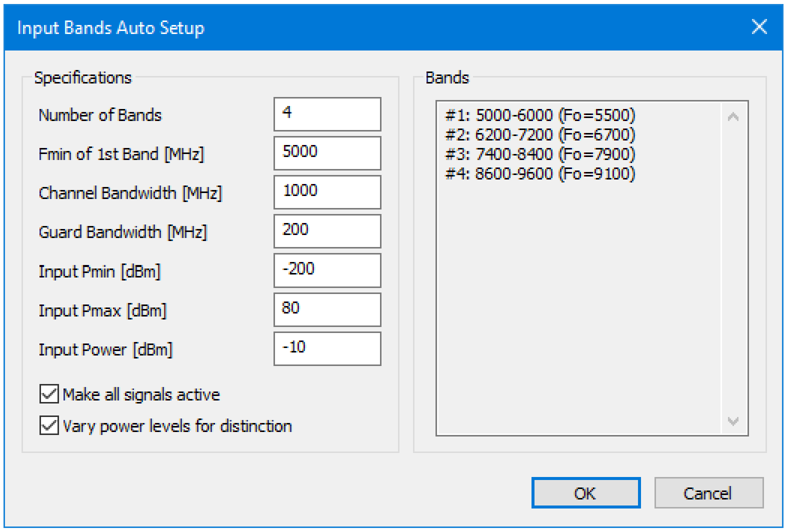
You can store input signals in an Input Signal Library and load from it as well. To add the current input bands to the library, click the **Add Set** button. To show the library click the **Show** button. When a library is shown and a signal set is selected, the name displays under **Input Signal Library**. When you edit signal properties, click the **Update** button to store the modified set back into the library.

The input signal library is stored in a text file in the User folder as *ifp_SysInputs.txt*. You can manually edit this file. If the file does not exist, click the **Add Set** button once to create the file and fill it with the current signal set. You can open the file and inspect for the format.

Input bands are displayed in graphical format at the bottom of the dialog box. The threat bands are drawn in red, and the normal (friendly) bands are drawn in blue. **Change Plot Palette** and **Copy Plot Image to Clipboard** buttons are provided to change the color palette and to copy the drawing onto the Clipboard as an image.

13.15.4.2. Input Bands Auto Setup Dialog Box

The Input Bands Auto Setup dialog box provides a quick way to set up input bands of the same widths and separated by identical gaps. To access this dialog box, click the **Auto Setup** button in the Input Signal Bands dialog box.



Number of Bands specifies the number of input frequency bands for a system.

Fmin of 1st Band is the lower corner of the first input band.

Channel Bandwidth is the bandwidth of each input band.

Guard Bandwidth is the separation between input bands. When this value is positive, the bands are separated from each other by this amount. When the value is negative, bands overlap, (the lower frequency of one band is lower than the upper frequency of the previous band by the guard bandwidth).

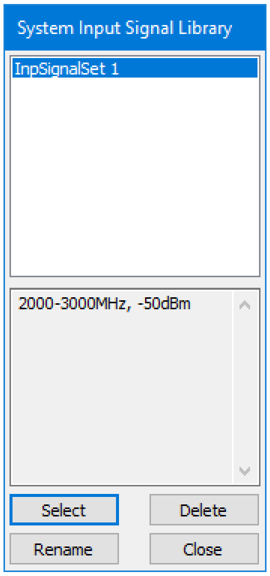
Input Pmin, **Input Pmax**, and **Input Power** are the minimum, maximum and nominal values of the input band power level.

The **Make all signals active** check box sets all of the input bands as active. When this check box is not selected, only the first input band is active; the other bands are disabled but ready for use.

The **Vary power levels for distinction** check box sets the nominal powers of input bands by decrementing 1dB. For example, if band 1 is 0dBm, band 2 is -1dBm, band 3 is -2dBm and so on.

13.15.4.3. System Input Signal Library Window

The System Input Signal Library window displays the input signal library and allows simple library operations. To access this window, click the **Show** button in the Input Signal Bands dialog box.



The list box at the top of the window lists the library items. Each item contains frequency bands and power levels as specified in the Input Signal Bands dialog box. When you select an item, its properties display in the lower half of the window.

Choose an item for use in the Input Signal Bands dialog box by selecting it and then clicking the **Select** button. In that dialog box under **Edit Bands**, the selected item shows "sel" appended to the band properties.

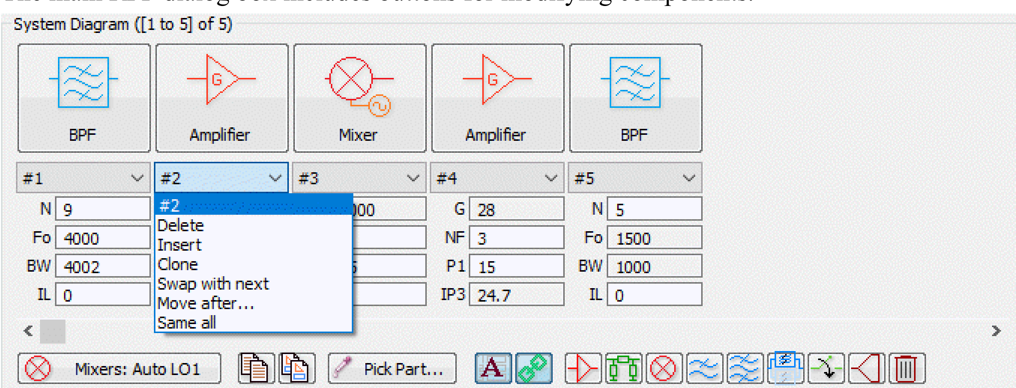
To delete a signal set from the library, click the **Delete** button.

To rename a signal set, click the **Replace** button.

To close the window, click the **Close** button.

13.15.5. Component Editing

The main RFP dialog box includes buttons for modifying components.



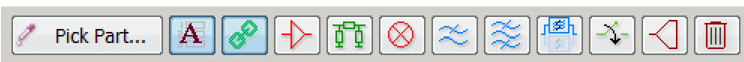
A drop-down box beneath each component button provides options for the following simple editing operations:

- Click **Delete** to delete the selected component.

- Click **Insert** to insert a component before the selected component. An Insert dialog box displays to allow you to select a position.
- Click **Clone** to insert a copy of the selected component directly before the component.
- Click **Swap** to exchange the selected component with the component that follows it.
- Click **Replace** to replace the selected component with a component from the library. This option is only available when the component has an associated, opened library.
- Click **Same all** to locate all components of the same kind and match their properties with those of the selected component. This command is useful, for example, when you change an amplifier in a cascade of amps and intend to replace all other amplifiers with that one.

13.15.5.1. Adding Component Shortcuts

The toolbar in the **System Diagram** group is used to add and modify system components.



To add a part from a library, click the **Pick Part** button. A Part Library window displays with a list of parts you can select.

When you click the **Add Components with Auto-Parameters** button, the system is in auto-parameter mode, where the auto-parameters of newly added components are set to Auto. For example, if you add a bandpass filter to the system in the auto-parameters mode, center frequency and bandwidth of the filter are set to Auto, so the system sets them automatically to pass the intended signal band.

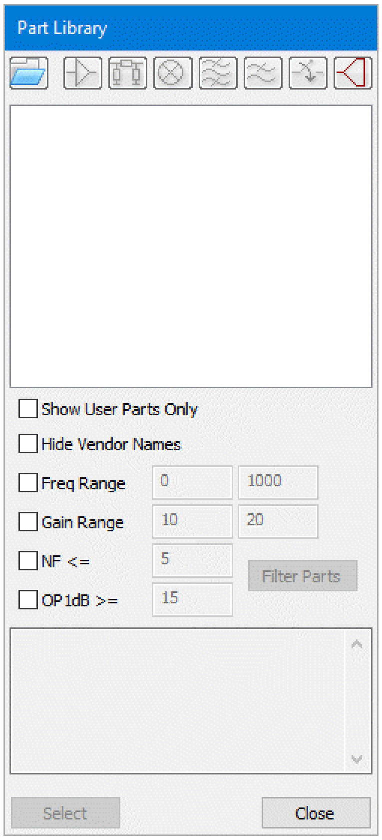
When you click the **Link Similar Parameters** button, the system links parameters of similar components. For example, if you change the noise figure of an amplifier, the noise figure of all amplifiers in the diagram are synchronized to the new value.

The remaining buttons are for adding amplifier (AMP), attenuator (ATT), mixer (MIX), lowpass filter (LPF), bandpass filter (BPF), switched bandpass filter (SBP), RF switch (SWT), and analog-to-digital converter (ADC) components.

When you click the **Clear Components** button all components in the system diagram are deleted.

13.15.5.2. Part Library Window

The Part Library window allows you to select parts from a system parts library. To access this window, click the **Pick Part** button on the toolbar in the main RFP dialog box. You can also access this dialog box by clicking the **Pick Part** button in the component Edit dialog boxes.



RFP uses two sets of libraries: Factory shipped libraries and User libraries. The file format is the same for both types. Factory shipped libraries are read-only and provided only for reference.

The recommended location for both libraries is the *Appdata* folder. The AWR Design Environment platform uses this folder to store many helper and data files. RFP also uses this location for its INI file, *ifp_Prefs.ini*.

Inside the RFP INI file there are two properties (Library and FactLib) that define the location of the Factory Library and the User Library:

```
[Folders]
Library= C:\Users\USERNAME\AppData\Local\AWR\Design Environment\RFP_Library_User
FactLib= C:\Users\ USERNAME \AppData\Local\AWR\Design Environment\RFP_Library_Factory
```

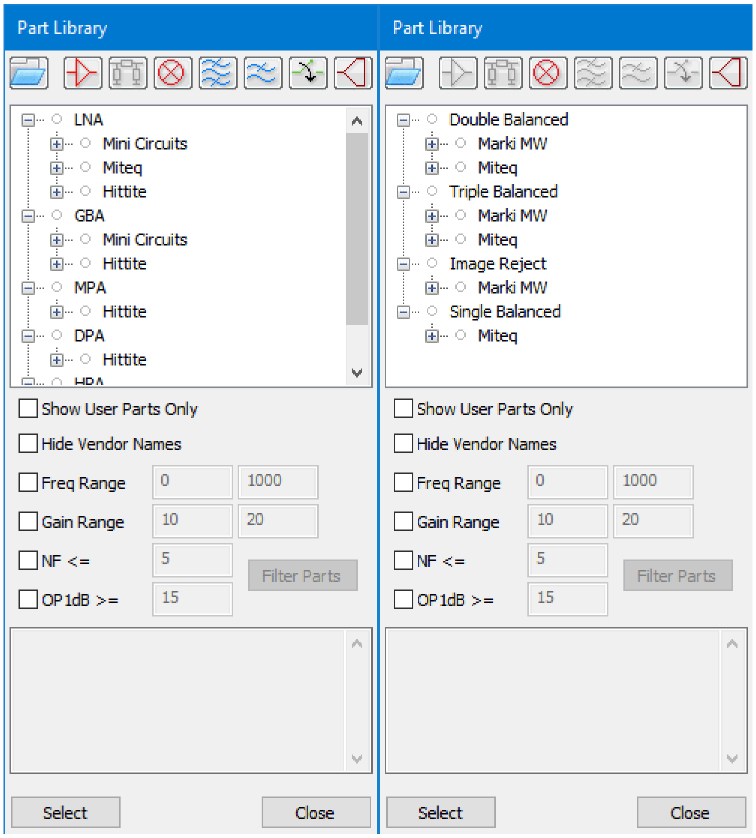
If the *ifp_Prefs.ini* file does not exist, run RFP and save an instance of the RFP Wizard to create the *.ini* file. You can edit this file at any time to update the 'Library' and 'FactLib' properties.

Create the *Library* and *FactLib* folders for the library data.

Download the RFP Factory Library by logging in to the Customer Portal on the Supports > Downloads page of the Cadence website. Select the **Component Libraries** tab, download the 'RFP Factory Library' file, then unzip the library files into the 'RFP_Library_Factory' directory you created.

Name	Date modified	Type
BFCN_1525p.s2p	5/22/2014 4:13 PM	S2P File
BFCN_1840p.s2p	5/22/2014 4:13 PM	S2P File
BFCN_2275p.s2p	5/22/2014 4:14 PM	S2P File
BFCN_2700p.s2p	5/22/2014 4:18 PM	S2P File
BFCN_2900p.s2p	5/22/2014 4:21 PM	S2P File
BFCN_3010p.s2p	5/22/2014 4:20 PM	S2P File
BFCN_4440p.s2p	5/22/2014 4:19 PM	S2P File
BFCN_7200p.s2p	5/22/2014 4:19 PM	S2P File
BFCN_7500p.s2p	5/22/2014 4:19 PM	S2P File
BFCN_8000p.s2p	5/22/2014 4:16 PM	S2P File
BFCN_8350p.s2p	5/22/2014 4:16 PM	S2P File
BFCN_8450p.s2p	5/22/2014 4:16 PM	S2P File
BFCN_8650p.s2p	5/22/2014 4:16 PM	S2P File
ifp_AMP.txt	5/23/2012 12:04 PM	Text Document
ifp_BPF.txt	6/18/2014 3:03 PM	Text Document
ifp_LPF.txt	6/18/2014 3:01 PM	Text Document
ifp_MIX.txt	1/15/2013 2:24 PM	Text Document
ifp_SWT.txt	5/23/2012 12:04 PM	Text Document
ifp_SysInputs.txt	10/20/2013 11:58 ...	Text Document
LFCN_80.s2p	5/22/2014 3:19 PM	S2P File
LFCN_120.s2p	5/22/2014 3:19 PM	S2P File
LFCN_225.s2p	6/18/2014 2:57 PM	S2P File
LFCN_400.s2p	6/18/2014 2:58 PM	S2P File
LFCN_1000.s2p	5/22/2014 3:20 PM	S2P File
LFCN_1200.s2p	5/22/2014 3:21 PM	S2P File
LFCN_1800.s2p	5/22/2014 3:23 PM	S2P File
LFCN_3000.s2p	5/22/2014 3:08 PM	S2P File
LFCN_5000.s2p	6/18/2014 2:58 PM	S2P File

To verify that the Factory Library is scanned and used by RFP, in the AWR Design Environment platform open the RFP Wizard and click the **Pick Part** button. In the following figure, the Part Library window on the left displays the available amplifier models. Click the Mixer symbol at the top of the window to list the available mixer models, as shown in the Part Library window on the right.



You can manually edit and store User libraries anywhere on the computer, however the recommended location is the *Appdata* folder. To change this location, click the **Set Folder for Part Libraries** button at the top left of the Part Library window, browse to another directory, then click **OK**.

The parts are listed with A and U icons referring to factory shipped (Cadence) and User types. To display only the User libraries, select the **Show User Parts Only** check box. To filter out the displayed parts based on their operating frequency range, select the **Frequency Range Filtering** check box, enter minimum and maximum values and click the **Apply** button.

Part libraries are stored in simple text files with an intuitive format that you can edit for custom or commercial parts. Each part library has a unique file name and format. In library files, each line corresponds to a part, and properties are separated by a comma. Text beyond the "!" character is ignored.

RFP currently uses the following part libraries:

Amplifier library: *ifp_AMP.txt*

The following shows the format and some sample data for this library. Note that the data is wrapped into several lines for display purposes; the actual file must have one line per part.

```
!Make           PartNo      Type   Fmin   Fmax   Gain   NF    OCP1 OIP3 Gslop Fgain
!
Mini Circuits, ZEL-0812LN, LNA,   800,   1200, 20.0, 1.5,   8.0, 18.0, 0.0,  0.0,
Mini Circuits, ZEL-1217LN, LNA,   1200, 1700, 20.0, 1.5, 10.0, 25.0, 0.0,  0.0,

Vdd    Idd    IVSWR OVSWR   Pkg
!
```

```
15.0, 70.0, 2.5,  2.5,   Conn
15.0, 70.0, 2.5,  2.5,   Conn
```

The `Make`, `PartNo`, and `Type` properties can be any text and are used for classification and displaying the part in the tree. `Fmin` and `Fmax` give the usable range of the part in MHz. `Gain`, `NF`, `OCP1`, and `OIP3` are budget parameters. `Gslop` is the gain slope [dB/GHz]; see "Edit AMP Dialog Box" for details on its use. `Fgain` is the frequency where `Gain` is defined. `Vdd` and `Idd` are supply parameters used to calculate total system power, which displays in the System Information window. `IVSWR` (input VSWR), `OVSWR` (output VSWR), and `Pkg` (package) data are reserved for future versions of the RFP Radio Frequency Planning Wizard.

Mixer library: *ifp_MIX.txt*

The following shows the format and some sample data for this library. Note that the data is wrapped into several lines for display purposes; the actual file must have one line per part.

```
!Make      PartNo    Type              RFmin   RFmax   LOmin   LOmax   IFmin   IFmax
!
Marki MW,  M1-0204,  Double Balanced,  2000,   4000,    2000,   4000,    0,    2000,
Marki MW,  M1-0208,  Double Balanced,  2000,   8000,    2000,   8000,    0,    2000,

LOPow    ConvL    SSBNF    IIP3     ICP1   Is(R/I)  Is(L/I)  Is(L/R)   ImgRj   RefPin   M   N
!
15.0,     5.0,     5.0,   16.0,     6.0,    20.0,    20.0,    38.0,     0.0,  -10.0,   5,  5,
15.0,     6.0,     6.0,   16.0,     6.0,    20.0,    20.0,    38.0,     0.0,  -10.0,   5,  5,

(Spurii suppressions: MxN=(1x1),(2x1),(3x1),...,(1x2),(2x2),(3x2),..  M must be equal to
 N,
 1x1 will be overridden to 0dBc)
!
(   0,  20,  10,  25,  20,  55,  50,  50,  50,  50,  50,  70,  55,  70,  60,  80,  95,

(   0,  20,  10,  25,  20,  55,  50,  50,  50,  50,  50,  70,  55,  70,  60,  80,  95,

!
95, 100,  90, 100, 100,  90, 110,  95),
95, 100,  90, 100, 100,  90, 110,  95),

RSWR  LSWR  ISWR  Pkg
!
2.5,  1.5,  1.0, SMT
2.5,  1.5,  1.0, SMT
```

The `Make`, `PartNo`, and `Type` properties can be any text an are used for classification and displaying the part in the tree. `RFmin`, `RFmax`, `LOmin`, `LOmax`, `IFmin`, and `IFmax` give the usable range of the part in MHz. `LOPow` (LO Power), `ConvL` (conversion loss), `SSBNF` (noise figure), `IIP3` (input IP3), and `ICP1` (input compression point) are parameters as shown in the "Edit MIX". `Is(R/I)`, `Is(L/I)`, and `Is(L/R)` are isolation in dB for RF/LO/IF leakages. `Is(R/I)` and `Is(L/I)` are used in the analysis as part of the Spur Table. `Is(L/R)` and `ImgRj` (image rejection) are reserved for future versions of the RFP RF Planning Tool Wizard. `RefPin` is the reference input power for the Mixer Spur Table. `M` and `N` are the dimensions of the Spur Table. The property in parentheses is a comma-separated Spur Table. Spur Table entries are given in positive numbers that correspond to suppression in dBc. The values are ordered by `N` first, and then `M`.

Example: 3,3 (0,5,10,15,20,25,30,35,40) is decoded as in (MxN) pairs as

```
(0x0)=0dBc,
(0x1)=Although it is specified here, it is overridden by L/I isolation.
(0x2)=10dBc
(1x0)= Although it is specified here, it is overridden by R/I isolation.
(1x1)=Although it is specified as 20dBc, it is overridden by RFP as 0 because this is the

 reference power.
(1x2)=25dBc
(2x0)=30dBc
(2x1)=35dBc
(2x2)=40dBc
```

RSWR (RF VSWR), LSWR (LO VSWR), ISWR (IF VSWR) and Pkg (package) are reserved for this version of the RFP RF Planning Tool Wizard.

RF Switch library: *ifp_SWT.txt*

The following shows the format and some sample data for this library. Note that the data is wrapped into several lines for display purposes; the actual file must have one line per part.

```
!Make      PartNo      Type   Fmin Fmax     IL      Isol     ICP1     IIP3     Vcont  Pkg
!
Hittite,   HMC190AMS8, SPDT,  1,   3000,    0.4,    30.0,    99.0,    99.0,     3.0 , MS8
Hittite,   HMC194MS8,  SPDT,  1,   3000,    0.7,    27.0,    30.0,    99.0,     5.0 , MS8
Hittite,   HMC197A,    SPDT,  1,   3000,    0.4,    50.0,    23.0,    99.0,     3.0 , SOT26
```

The `Make`, `PartNo`, and `Type` properties can be any text and are used for classification and displaying the part in the tree. `Fmin` and `Fmax` give the usable range of the part in MHz. `IL`, `ICP1`, and `IIP3` are budget parameters. `Isol` is the isolation in dB when the switch is off. `Vcont` (control voltage) and `Pkg` (package) are reserved for future versions of the RFP Radio Frequency Planning Wizard.

RF Lowpass Filter library: *ifp_LPF.txt*

The following shows the format and some sample data for this library. Note that the data is wrapped into several lines for display purposes; the actual file must have one line per part.

```
!Make            PartNo          Type        Order   Fc      Datafile
FiltCompany1,     LP111,        Lumped,      5,      400,    Filter11.s2p
FiltCompany2,     LP121,        SSS,         7,      500,    Filter22.s2p
FiltCompany1,     LP131,        Cavity,      9,      600,    Filter33.s2p
```

The `Make`, `PartNo`, and `Type` properties can be any text and are used for classification and displaying the part in the tree. `Order` represents the degree of the filter, `Fc` is the passband cutoff frequency, and `Datafile` is the file name from which the S-parameters are read. The S-parameter file must exist in the User Library folder.

RF Bandpass Filter library: *ifp_BPF.txt*

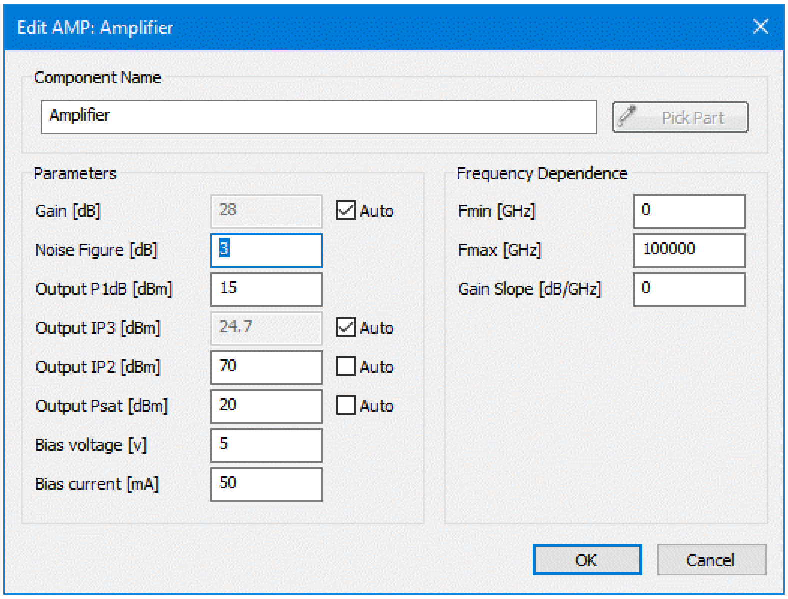
The following shows the format and some sample data for this library. Note that the data is wrapped into several lines for display purposes; the actual file must have one line per part.

```
!Make            PartNo          Type        Order   Fmin     Fmax    Datafile
FiltCompany1,     BP111,         Lumped,     5,      400,     600,    Filter1.s2p
FiltCompany2,     BP121,         SSS,        7,      400,     600,    Filter2.s2p
FiltCompany1,     BP131,         Cavity,     9,      400,     600,    Filter3.s2p
```

The `Make`, `PartNo`, and `Type` properties can be any text and are used for classification and displaying the part in the tree. `Order` represents the degree of the filter, `Fmin` and `Fmax` are the passband corners, and `Datafile` is the file name from which the S-parameters are read. The S-parameter file must exist in the User Library folder.

13.15.5.3. Edit AMP Dialog Box

The Edit AMP dialog box is used to edit the parameters of an amplifier. To access this dialog box, click an Amplifier component button in the **System Diagram** section of the main RFP dialog box.



Component Name is the name loaded from the library file when you select a part from the library.

Gain is the nominal gain of the amplifier. When you set it to auto by selecting the **Auto** check box, its value is calculated by the system to achieve the overall gain for the target specification.

Noise Figure, **Output P1dB**, **Output IP3**, and **Output IP2** are all intuitive parameters. When set to **Auto**, **Output IP3** and **Output IP2** are calculated as follows:

```
OIP3 = OP1dB + 9.7
OIP2 = OP1dB + 20
```

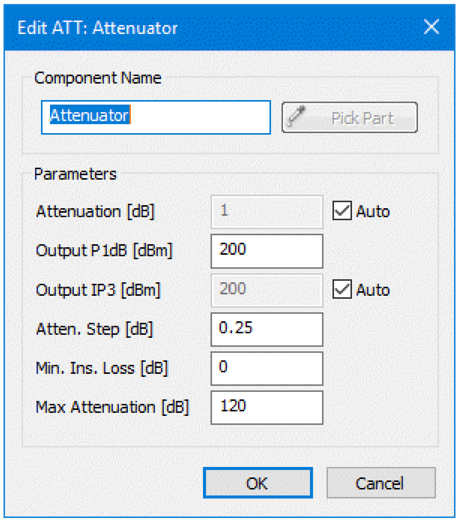
Bias voltage and **Bias current** are supply parameters used to calculate total system power, which is calculated and displayed in the System Information window.

In the **Frequency Dependence** group, minimum and maximum frequency range for the component is specified. Outside the frequency range specified, the gain of the amplifier is changed by the **Gain Slope**. The equations that govern the gain variation at frequency F are given as:

```
G = Gnom – S * (Fmin – F) for F < Fmin
G = Gnom for Fmin < F < Fmax
G = Gnom – S * (F – Fmax) for F < Fmax
```

13.15.5.4. Edit ATT Dialog Box

The Edit ATT dialog box is used to edit the parameters of an attenuator. To access this dialog box, click an Attenuator component button in the **System Diagram** section of the main RFP dialog box.



Component Name is the name loaded from the library file when you select a part from the library.

Attenuation is the set-attenuation of the attenuator. This is the fixed attenuator value for passive attenuators. When you set it to auto by selecting the **Auto** check box, its value is calculated by the system to achieve the overall gain for the target specification.

Output P1dB and **Output IP3** are output compression point and third-order output intercept points. When you set it to auto by selecting the **Auto** check box, output IP3 is calculated as follows:

```
OIP3 = OP1dB + 9.7
```

Atten. Step is used in the main RFP dialog box when you scroll your mouse wheel over the attenuation to increase or decrease the attenuation step.

Min. Ins. Loss is a fixed loss associated with the component. For digital attenuators, this is the insertion loss in the data sheet when the attenuator is set to 0dB. Total attenuation for the component is therefore the sum of attenuation and the minimum insertion loss.

Max Attenuation is the upper limit of the attenuation and is mainly used for digital attenuators.

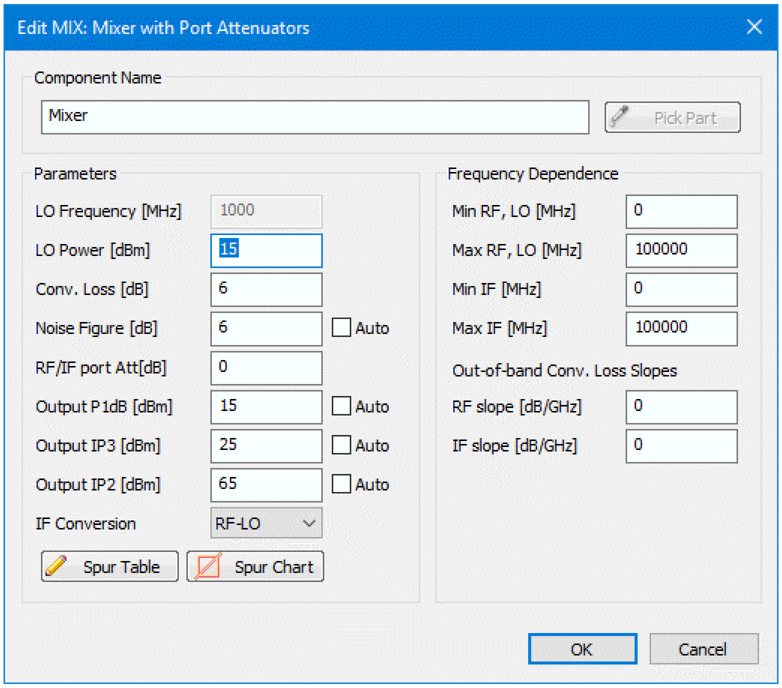
Example: For a 4-bit 15dB digital attenuator with 1.2dB insertion loss at its 0dB state, set the parameters as the following:

```
Attenuation = 0
Attenuation Step = 1
Minimum Insertion Loss = 1.2
Maximum Attenuation = 15
```

Then in the **System Diagram** section of the main RFP dialog box, you can scroll your mouse wheel over the attenuation to set it within the operating range of the component.

13.15.5.5. Edit MIX

The Edit MIX dialog box is used to edit the parameters of a mixer with input and output attenuators. To access this dialog box, click a Mixer component button in the **System Diagram** section of the main RFP dialog box.



Component Nameis the name loaded from the library file when you select a part from the library.

LO Frequency and **LO Power** are the local oscillator properties of the mixer. **LO Frequency** is only available for editing if the conversion scheme allows it.

Conv. Loss is the RF to IF conversion loss of the mixer. For most passive mixers, the value ranges from 6.5 to 8.

RF/IF port Att is the value of attenuators at both RF and IF ports of the mixer. This parameter is provided to save space in the system diagram. Alternatively, you can use individual attenuators by setting the parameter to 0.

Output P1dB, **Output IP3**, and **Output IP2** are output compression point and third and second order output intercept points. When set to **Auto**, they are calculated as follows:

```
OP1dB = LO Power - 5
OIP3 = OP1dB + 9.7
OIP2 = OP1dB + 20
```

IF Conversion is the intended conversion scheme for this mixer. Three options are available:

- IF = RF – LO
- IF = RF + LO
- IF = LO – RF

In some cases, the difference function (RF-LO or LO-RF) produces (-) frequency values due to improper selection of LO and RF inputs. In these cases, the program tries to use the alternative difference function momentarily to make the output frequency positive.

The **Frequency Dependence** group provides a mechanism to simulate out-of-band behavior of the mixer. You can specify RF and IF ranges through **MinRF(LO)**, **MaxRF(LO)**, **MinIF** and **MaxIF** parameters. Conversion loss slopes are specified by **RF slope** and **IF slope**. Outside the specified frequencies, the slope S modifies the conversion loss (CL) as follows:

```
CL = CL − S * (Fmin − F)   for F < Fmin
CL = CL         for Fmin <F < Fmax
CL = CL − S * (F − Fmax)   for F > Fmax
```

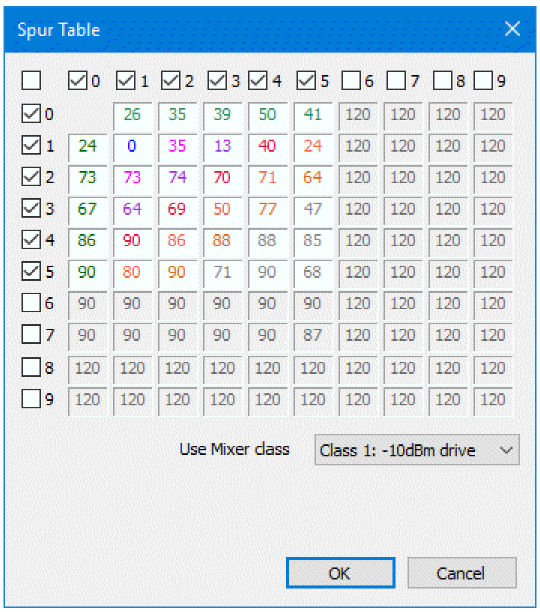
Since there are two slope parameters, the effect is additive. You can set either RF or IF or both ranges and slope to simulate out-of-band conversion of the mixer.

To edit the spur suppression of the mixer, click the **Spur Table** button. A Spur Table dialog box displays for editing the 9x9 Spur Table.

To edit the spur suppression of the mixer and simulate a single stage conversion chart, click the **Spur Chart** button. A Spur Chart dialog box displays to enable editing the Spur Table.

13.15.5.6. Spur Table Dialog Box

The Spur Table dialog box is used to edit the spur suppression table of a mixer. To access this dialog box, click the **Spur Table** button in the "Edit MIX".



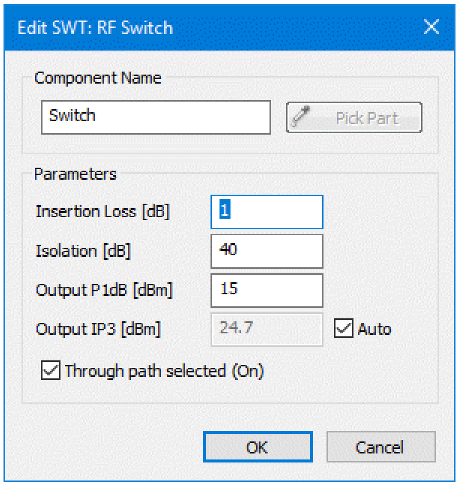
You can enable editing and analysis of table rows and columns by selecting the check boxes next to M,N numbers. Spur Table rows belong to M (RF), and columns belong to N (LO). For example, the horizontal cells and check box (4) belong to the suppression m=4 in the equation IF = m*RF + n*LO. Similarly, the vertical cells and check box (4) belong to the suppression n=4 in the equation IF = m*RF + n*LO.

When cells are greyed, they are not editable and are not used in the analysis. Cells are color coded. The spur plots or spectrum plots use the same color codes as the table entries. Blue represents the intended 1x1 component as seen in the table. In the spur and spectrum plots, the intended signal is also drawn in blue.

The **Use Mixer class** drop-down allows quick setting of the table from standard double balanced mixer classes. The drive level in the selected option is used as a reference input power for the mixer.

13.15.5.7. Edit SWT Dialog Box

The Edit SWT dialog box is used to edit the parameters of an RF switch. To access this dialog box, click a Switch component button in the **System Diagram** section of the main RFP dialog box.



Component Name is the name loaded from the library file when you select a part from the library.

Insertion Loss is the loss of the switch in the ON state.

Isolation is the loss of the switch in the OFF state.

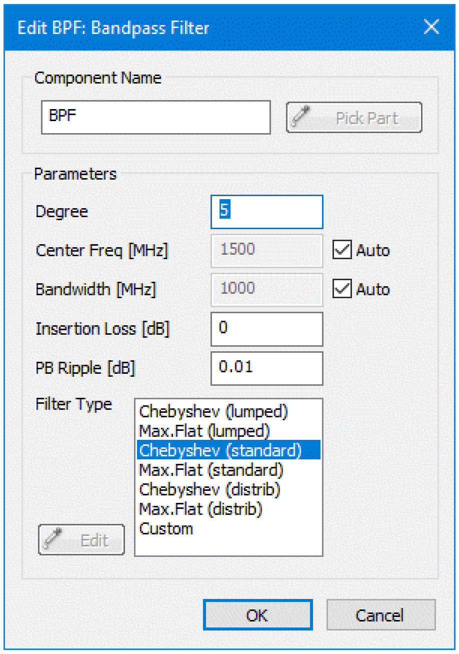
Output P1dB and **Output IP3** are output compression point and third-order output intercept points. When set to **Auto**, output IP3 is calculated as:

```
OIP3 = OP1dB + 9.7
```

The **Through path selected (On)** check box sets the switch ON or OFF. When selected, the switch is selected ON (through path).

13.15.5.8. Edit BPF Dialog Box

The Edit BPF dialog box is used to edit the parameters of a bandpass filter. To access this dialog box, click a BPF component button in the **System Diagram** section of the main RFP dialog box.



Component Name is the name loaded from the library file when you select a part from the library.

Degree is the prototype order of the bandpass filter. It is also equal to the number of resonators for microwave filters.

Center Freq and **Bandwidth** are major passband parameters for bandpass filters. When you set **Center Freq** to **Auto**, the program automatically sets it to where the intended signal center or IF center is. When you set **Bandwidth** to **Auto**, the bandwidth is automatically set wide enough to allow the desired frequency range (or converted IF range) to go through. For example, if RF=5000-6000 is input to a mixer with LO=4000 and the conversion scheme is RF-LO, a bandpass filter used at the mixer output automatically sets itself to 1000-2000 to allow the desired IF to pass.

Insertion Loss is the loss for the whole passband.

PB Ripple is the passband ripple for Chebyshev filter types. Together with **Degree**, ripple helps determine the attenuation of a filter outside its passband. Due to its diminishing value in analysis, it is ignored for frequencies that fall in the passband of the filter.

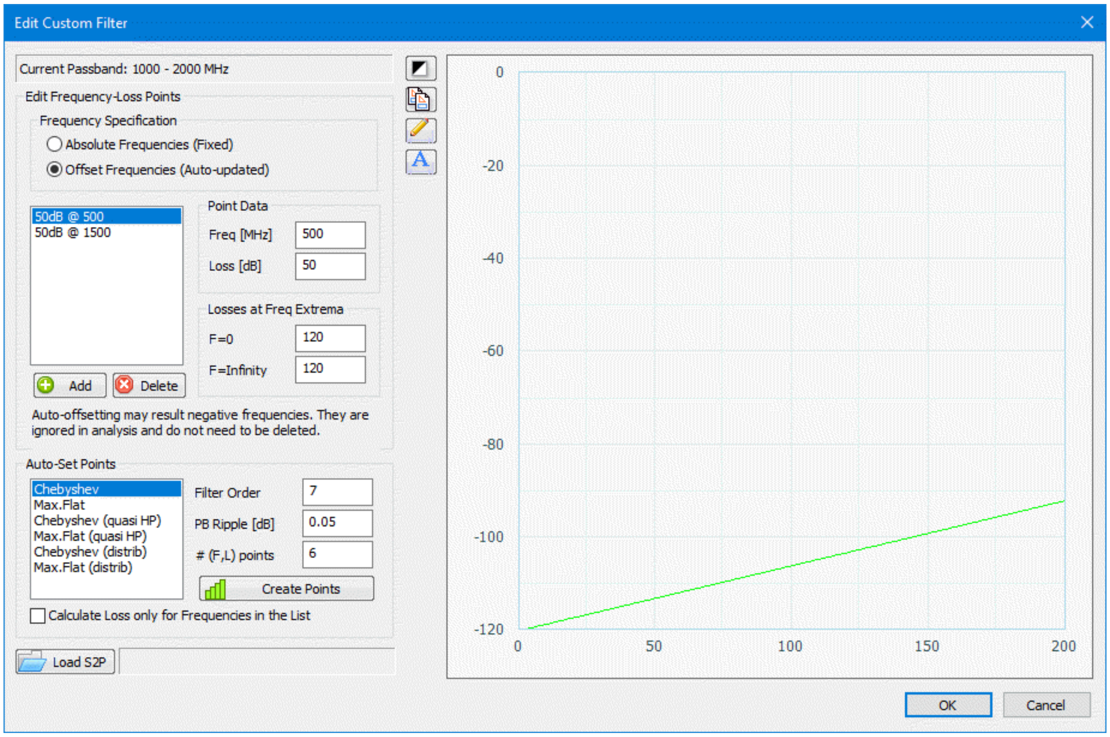
Filter Type lists the filter types available. There are mainly Chebyshev and Maximally Flat types with three frequency mapping options: Standard, Quasi HP, and Distributed. You can add a bandpass filter to a system and compare filter responses by changing the filter type.

When you set **Filter Type** to **Custom**, the **Edit** button is enabled to allow custom frequency-loss editing for the filter in the Edit Custom Filter dialog box.

When the design is generated in the VSS program, the **Custom** filter type is mapped to the AMP_B component with gain and frequency data set in its GAIN and FREQS parameters. **Chebyshev** (standard) and **Max. Flat** (standard) filter types are mapped to BPFC and BPFB respectively. Other filter types are mapped to LIN_S, where the S2P data is stored in the Project Browser as a separate file.

13.15.5.9. Edit Custom Filter Dialog Box

The Edit Custom Filter dialog box is used to edit parameters for a custom lowpass/bandpass filter. To access this dialog box, select **Custom** as the **Filter Type** and then click the **Edit** button in the "Edit BPF Dialog Box" or "Edit LPF Dialog Box".



At the top of the dialog box, the current passband frequencies display.

The **Frequency Specification** group specifies how the custom filter frequency points are interpreted. For **Absolute Frequencies**, the frequency points are assumed fixed and they are not changed by the program. For **Offset Frequencies**, the frequency points are updated as the reference frequency needs changing due to a changing input signal range. If, for example, the custom filter is a bandpass filter that follows a mixer, when the LO frequency changes, the IF center also changes. When **Offset Frequencies** is selected, this custom filter moves all frequency points accordingly so that the filter center frequency corresponds to the IF center and the shape of the filter is still preserved.

The list box in the middle of the dialog box displays frequency-loss (F,L) points of the filter. You can edit each (F,L) by changing the **Freq** and **Loss** under **Point Data**. Click elsewhere in the dialog box to update the data. You can also click in an option and scroll the mouse wheel to increment/decrement the values. The selected data point displays with a circle in the response plot.

The **Losses at Freq Extrema** group contains the loss parameters [dB] at **F=0** and **F=infinity**. These frequency extrema do not have to be added to the list, as they are internally added to the analysis.

To add a new data point, click the **Add** button.

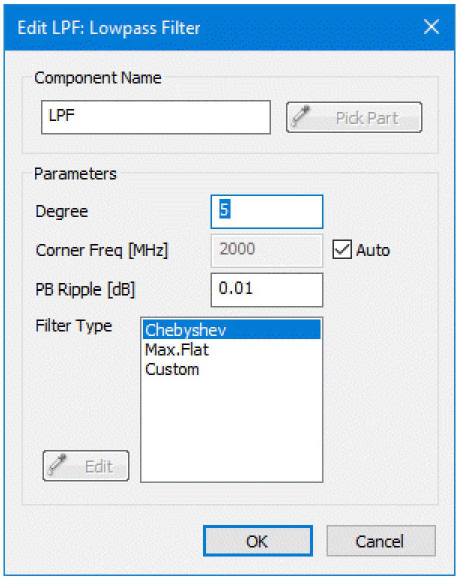
To delete the selected data point, click the **Delete** button.

Custom (F,L) data is used by interpolation. During analysis, if a frequency point falls between two data points, the attenuation of the filter is calculated by using a linear approximation.

The **Auto-Set Points** group contains options for quickly setting up the (F,L) data with **Filter Order**, **PB Ripple**, and **# (F,L) points** options. By clicking the **Create Points** button, you can create the exact attenuation data for the given filter type for frequencies that are program-selected. Frequencies are estimated by the program depending on the bandwidth and number of points. If the current frequency values are good and only a new attenuation profile is desired, then select the **Calculate Loss only for Frequencies in the list** check box.

13.15.5.10. Edit LPF Dialog Box

The Edit LPF dialog box is used to edit the parameters of a lowpass filter. To access this dialog box, click an LPF component button in the **System Diagram** section of the main RFP dialog box.



Component Name is the name loaded from the library file when you select a part from the library.

Degree is the order of the lowpass filter.

Corner Freq is the cutoff frequency of the filter. For maximally flat filters, it corresponds to 3dB point. For Chebyshev filters, it corresponds to the ripple corner. You can set this option to **Auto** to allow the desired frequency range to go through.

PB Ripple is the passband ripple for Chebyshev filter types. Together with **Degree**, this option helps determine the attenuation of a filter outside its passband. Due to its diminishing value in analysis, it is ignored for frequencies that fall in the passband of the filter.

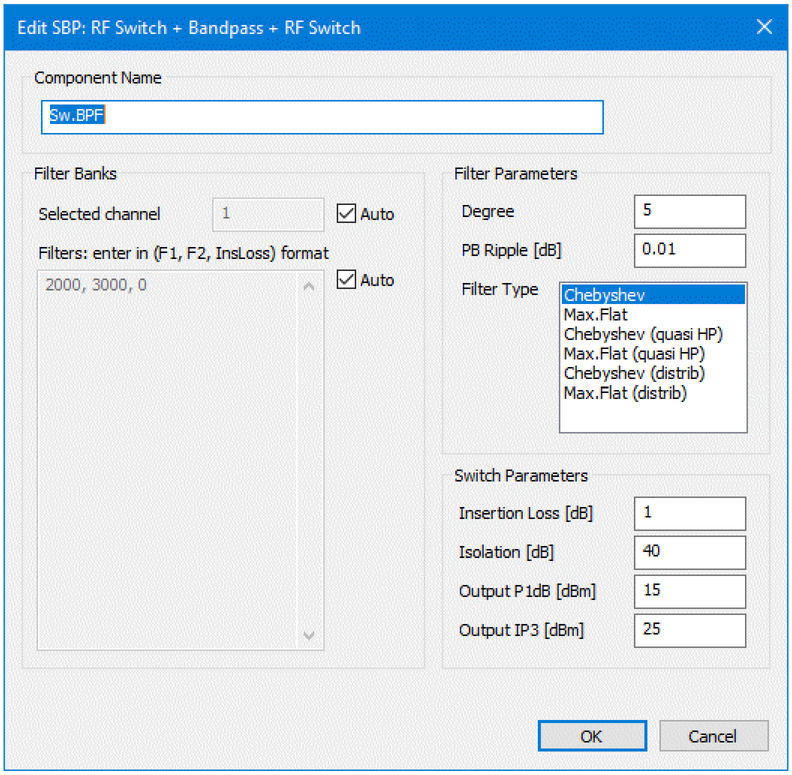
Filter Type lists the filter types available. When this option is set as **Custom**, the **Edit** button is enabled to allow custom frequency-loss editing for the filter in the Edit Custom Filter dialog box.

When the design is generated in the VSS program, the **Custom** filter type is mapped to the AMP_B component with gain and frequency data set in its GAIN and FREQS parameters. **Chebyshev** (standard) and **Max. Flat** (standard) filter types are mapped to LPFC and LPFB respectively.

13.15.5.11. Edit SBP Dialog Box

The Edit SBP dialog box is used to edit the parameters of a switched bandpass filter. To access this dialog box, click an Sw.BPF component button in the **System Diagram** section of the main RFP dialog box.

A switched bandpass filter contains an RF switch, followed by a list of filter channels with one actively selected, followed by another RF switch. It is provided for convenience and to save space in the system diagram. You can use individual switches and auto-set bandpass filters if preferred.



Component Name is the name loaded from the library file when you select a part from the library.

In the **Filter Banks** group, the bandpass filter channels and the **Selected channel** display. The channels display in (F1, F2, InsLoss) format. For example, 4000,4250,1 corresponds to a bandpass filter in the 4000-4250MHz range with an insertion loss of 1dB. The **Selected channel** shows the selected bandpass filter index in the list.

The **Filter Parameters** group contains options for the common parameters of filter channels.

Degree is the prototype order of the bandpass filter. It is also equal to the number of resonators for microwave filters

PB Ripple is the passband ripple for Chebyshev filter types. With **Degree**, **PB Ripple** helps determine the attenuation of a filter outside its passband. Due to its diminishing value in analysis, it is ignored for frequencies that fall in the passband of the filter.

Filter Type displays a list of filter types available. There are mainly Chebyshev and Maximally Flat types with three frequency mapping options: standard, quasi HP and distributed. You can add a bandpass filter to a system and compare filter responses by changing the filter type.

Degree, **PB Ripple**, and **Filter Type** are bandpass filter properties that are identical for all channels.

The **Switch Parameters** group contains the parameters of the input and output RF switches.

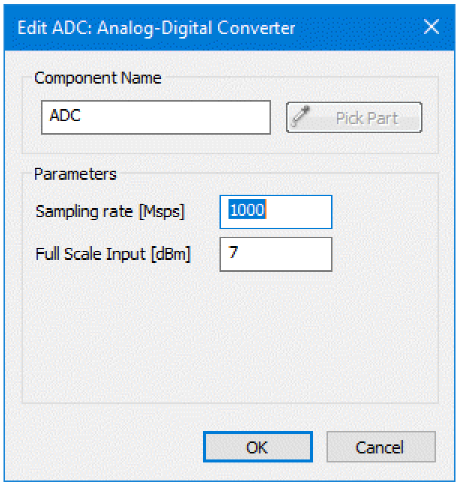
Insertion Loss is the loss of the switches. There are two switches in SBP, one at the input and one at the output. Therefore, the overall insertion loss of SBP is 2*Switch IL + Selected channel IL.

Isolation is the loss of the switches. This parameter is reserved for future use.

Output P1dB and **Output IP3** are output compression point and third-order output intercept points used when system budget is calculated.

13.15.5.12. Edit ADC Dialog Box

The Edit ADC dialog box is used to edit the parameters of an analog-to-digital converter. To access this dialog box, click an ADC component button in the **System Diagram** section of the main RFP dialog box.

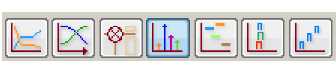


Component Name is the name loaded from the library file when you select a part from the library.

Sampling rate is the analog-to-digital sampling rate in Mega samples per second.

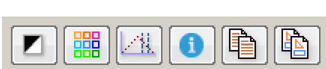
13.15.6. Viewing System Response

There are four groups of buttons on the System Response toolbar.



The first group of buttons change response viewing mode. From left to right, the buttons are for:

- Viewing the System Budget response
- Viewing the System Budget response with Sweep Parameter
- Viewing Spot Frequency Spur Schematic
- Viewing Spot Frequency Response
- Viewing Frequency Band Response
- Viewing Spot Frequency/Frequency Band Response for all systems in the design
- Viewing Spot Frequency/Frequency Band Response for the conversion stages in the selected system



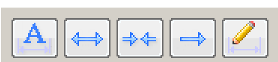
The second group of buttons provide various options for the system response. From left to right, the buttons are for:

- Changing the color palette of the drawing/graph

- Changing the data pattern of the drawing/graph (click to experiment)
- Showing/Hiding Nyquist zones. When shown, the dashed triangles show the Nyquist zones as determined by the ADC sampling frequency. The peak of the first triangle is FS/2. The vertical dashed lines show the actual input frequency band.
- Showing/Hiding plot information in the System Response Window
- Copying the System Response information to the Clipboard as text
- Copying the drawing/graph to the Clipboard as an image

The third group of buttons provide Y-axis scaling for graphs. From left to right, the buttons are for:

- Changing the Y-axes scale/div. Toggles between 1, 2, 5, 10, and 20dB per division.
- Increasing the Y-axis reference level. Reference level is the value on top of the Y-axis.
- Decreasing the Y-axis reference level. Reference level is the value on top of the Y-axis.
- Editing Y-axis properties. This option is only available in the System Budget Response mode.

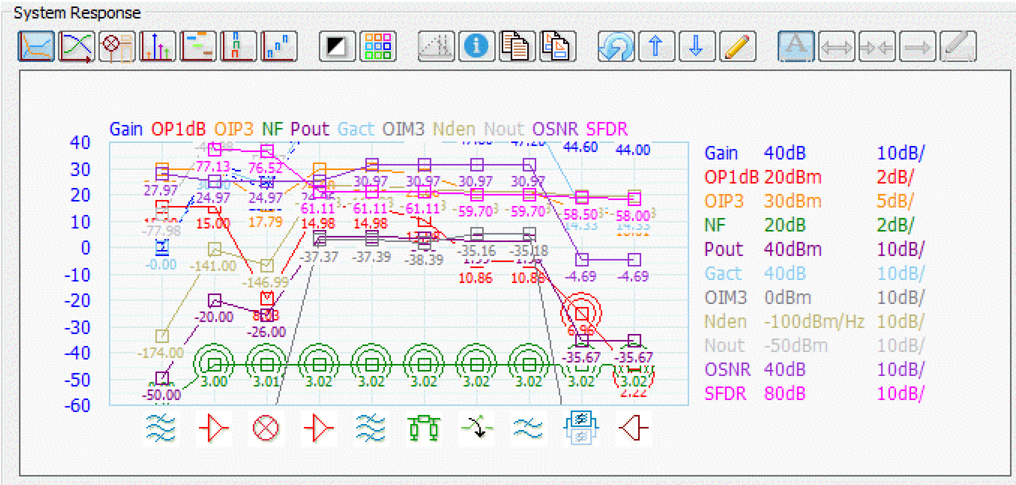


The fourth group of buttons provide X-axis (frequency) scaling for graphs. From left to right, the buttons are for:

- Setting the analysis frequency range to Auto.
- Increasing the analysis frequency span.
- Decreasing the analysis frequency span.
- Increasing the maximum analysis frequency.
- Editing the analysis frequency span. This option is only available when the frequency range is not Auto.

13.15.6.1. Budget Response

System Budget Response viewing modes plot various parameters of popular system budget calculations on a conveniently laid out graph. To select this mode, click the **Show Budget Response** button on the **System Response** group toolbar. The X-axis displays system components from left to right, and data traces show how the budget parameters change after each stage.



Y-axis scaling, the line color, and the legend all display in the same color as a parameter for easy distinction. On the right, the data legend shows the reference level and scale/div. for parameters. The Y-axis scaling on the left is given for the selected parameter. To select a different parameter, click the parameter in the legend. The traces do not move in the plot; the Y-axis values are updated to reflect selected parameter reference levels and scale/divs.

When a specific parameter misses the target system specs (for example Gain or NF) at a system stage, the value that corresponds to the violation point is circled twice as a warning. It is useful for parameters such as NF, which cannot be improved by further stages once it falls out of spec at any stage.

You can change the text values and the trace types by clicking the **Change Data Pattern** button.

To display the parameter values, click the **i** (Information) button to display the System Response window, which provides a comprehensive list of budget parameters calculated at the output of each stage. A legend displays at the bottom of the window.

System Response

2	AMP	31.30	2.09	15.65	25.35	70.64	-15.65	-5.95	39.34	-106.80	-140.61	25.89	77.31
3	MIX	25.95	2.09	9.19	18.95	59.45	-16.76	-7.00	33.50	-110.06	-145.96	25.88	76.61
4	AMP	56.60	2.10	15.63	25.33	69.77	-40.97	-31.27	13.17	-30.87	-115.30	25.88	60.42
5	BPF	57.25	2.10	16.28	25.98	70.16	-40.97	-31.27	12.91	-30.22	-114.65	31.90	60.42
6	ATT	56.90	2.10	15.93	25.63	69.56	-40.97	-31.27	12.66	-30.57	-115.00	31.90	60.42
7	SWT	56.55	2.10	12.61	22.31	68.98	-43.94	-34.24	12.43	-24.97	-115.35	31.90	58.44
8	LPF	57.20	2.10	13.26	22.96	69.39	-43.94	-34.24	12.19	-24.32	-114.70	31.90	58.44
9	SBP	55.85	2.10	9.00	18.84	67.40	-46.85	-37.01	11.55	-20.14	-116.05	31.90	56.59
10	ADC	56.50	2.10	3.46	18.34	14.33	-53.04	-38.16	-42.17	-21.27	-115.40	30.53	55.82

At T=MAX

#	After	Gain	NFig	OP1dB	OIP3	OIP2	IP1dB	IIP3	IIP2	OIM3	Nden	OSNR	SFDR
1	BPF	-0.60	0.60	14.40	24.10	69.39	100.00	100.00	100.00	-200.00	-174.00	27.38	148.93
2	AMP	28.80	4.20	14.40	24.10	69.39	-14.40	-4.70	40.59	-111.80	-141.00	23.78	76.73
3	MIX	22.20	4.21	6.94	16.69	57.54	-15.26	-5.51	35.34	-116.79	-147.59	23.77	76.19
4	AMP	51.60	4.22	14.37	24.07	68.32	-37.23	-27.53	16.72	-43.35	-118.18	23.76	61.50
5	BPF	51.00	4.22	13.77	23.47	67.49	-37.23	-27.53	16.49	-43.95	-118.78	29.78	61.50
6	ATT	49.40	4.22	12.17	21.87	65.71	-37.23	-27.53	16.31	-45.55	-120.38	29.78	61.50
7	SWT	47.80	4.22	9.07	18.77	63.96	-38.73	-29.03	16.16	-44.13	-121.98	29.78	60.50
8	LPF	47.20	4.22	8.47	18.17	63.23	-38.73	-29.03	16.03	-44.73	-122.58	29.78	60.50
9	SBP	44.60	4.22	4.67	14.44	60.31	-39.93	-30.16	15.71	-45.09	-125.18	29.78	59.75
10	ADC	44.00	4.22	0.71	13.39	13.06	-43.29	-30.61	-30.94	-44.77	-125.78	29.78	59.44

OIM3: 3rd order intermodulation level [dBm] at the output of each stage
Nden: noise density [dBc/Hz] at the output of each stage
OSNR: output signal to noise ratio [dB] at the output of each stage
Fcont: noise factor contributed by each stage

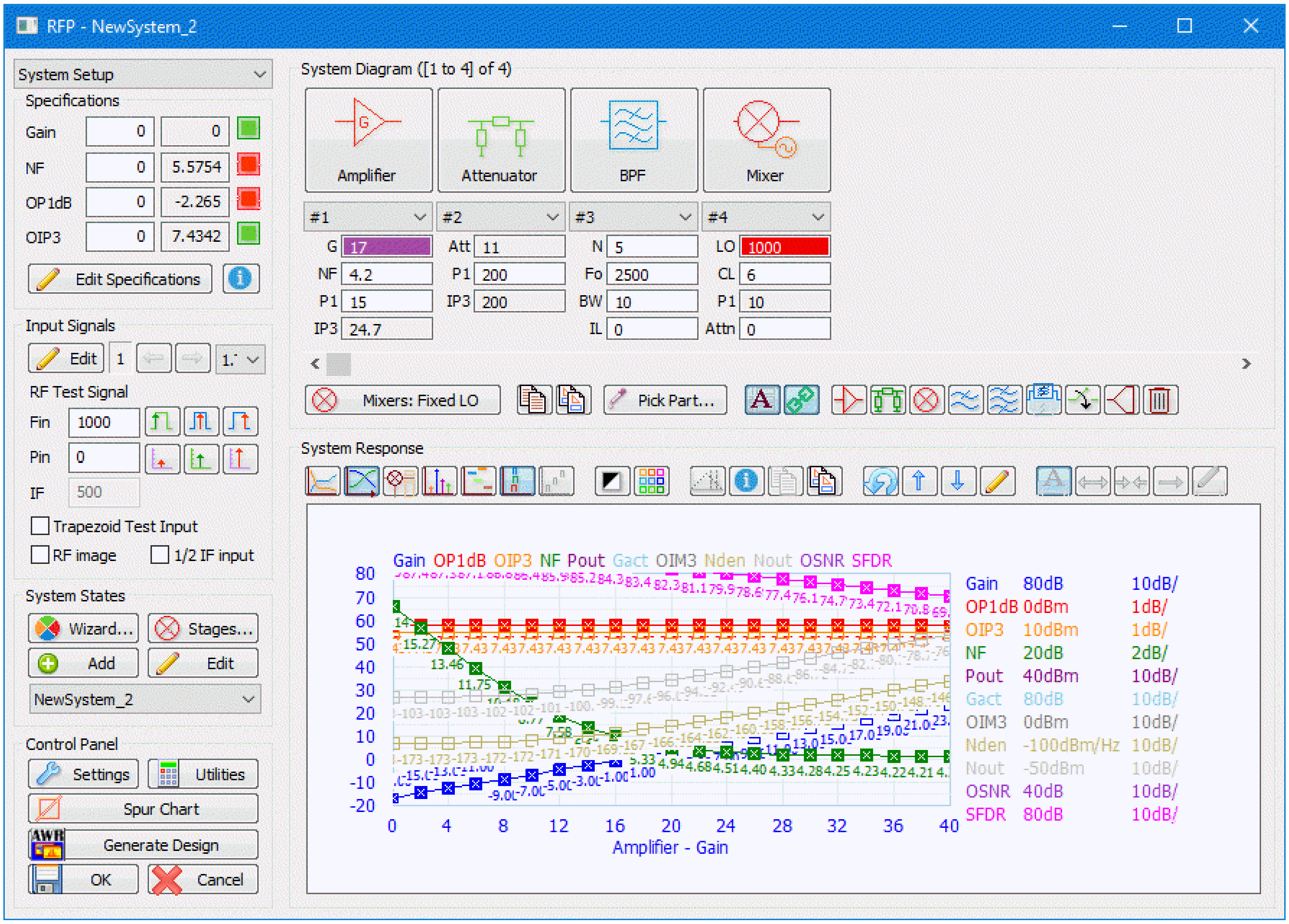
13.15.6.2. Budget Response with Sweep Parameter

You can view Budget Response by using a component parameter as a sweep variable. This mode is very useful when seeking an optimum value of a parameter. For example, what is the minimum LNA gain needed to maintain a specified NF? Although NF, P1dB are mostly intuitive, specifications such as SFDR are difficult to predict. In this view mode, you can plot SFDR against a component parameter with one mouse click.

To select this mode, click the **Show Budget Response with Sweep Parameter** button on the System Response toolbar.

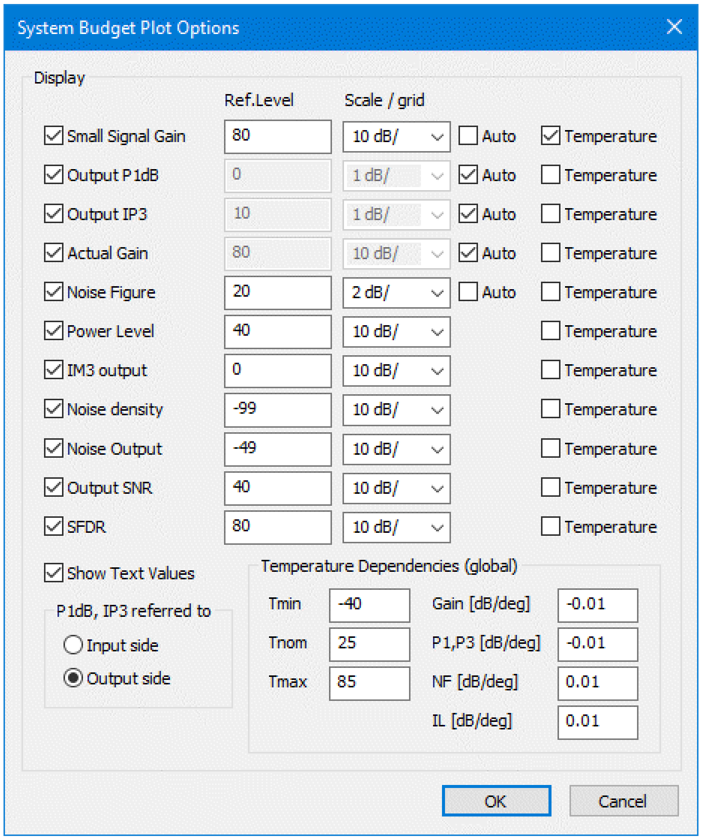
RFP now varies the selected component parameter and calculates overall system responses, then uses the parameter values as the X-axis and plots the responses. The selected parameter is a different color, as shown in the following figure.

Also shown in the figure is the amplifier's gain parameter selected (clicked on). RFP then varies the selected gain between 0 and 40dB and plots the overall Gain, NF and SFDR. The variation ranges are predefined and do not need specifying.



13.15.6.3. System Budget Plot Options Dialog Box

The System Budget Plot Options dialog box allows you to edit trace properties of the system budget plot. To access this dialog box, click the **Edit Left-axis scale/div** button on the **System Response** group toolbar.



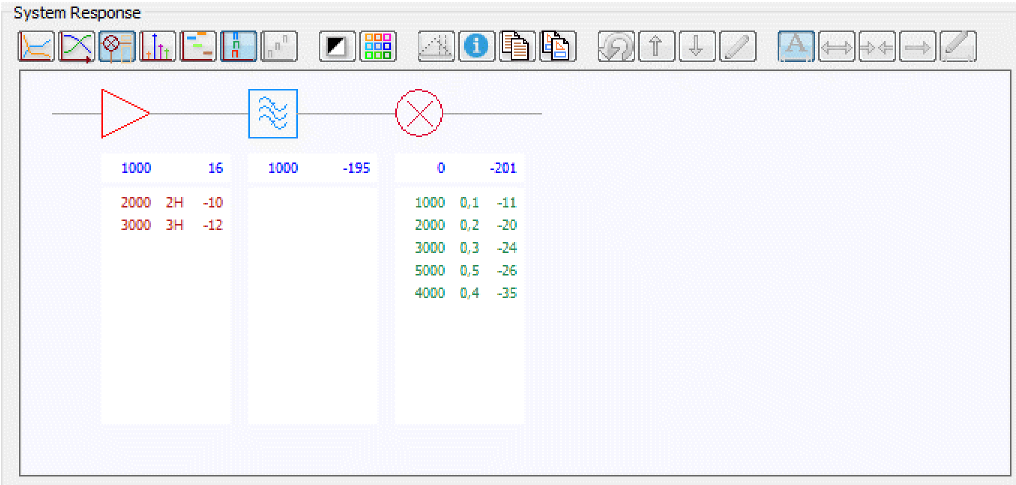
This dialog box contains all of the parameters you can plot. When you select the check box associated with a parameter, the parameter is plotted in the graph as a trace. You can select the Reference level and scaling for each parameter individually, and set the **Small Signal Gain**, **P1dB**, **IP3**, **Actual Gain**, and **Noise Figure** parameters to **Auto**. When set to auto, the reference level and scale/div of **Small Signal Gain** are used for these traces.

There is a difference between **Small Signal Gain** and **Actual Gain** parameters. **Small Signal Gain** is the cascaded calculation of gain and attenuation values that are specified for components, while **Actual Gain** is the gain calculated for the given input power. Since the gain may be compressed after stages, the actual gain may come out lower than expected. As the input power is decreased, actual gain approaches the small signal gain.

You can select **P1dB** and **IP3** as Input or Output parameters when plotting by selecting the **Input side** or **Output side** option at the bottom right of the dialog box. Output values are calculated by adding Small to the Input values for P1dB and IP3.

13.15.6.4. Spot Freq Schematic View Mode

Spot Frequency Schematic is a view mode that provides a simplified schematic of the system with spot frequency progression through stages. To save space, only frequency-contributing stages display. To select this mode, click the **Show Spot Freq Schematic** button on the System Response toolbar.

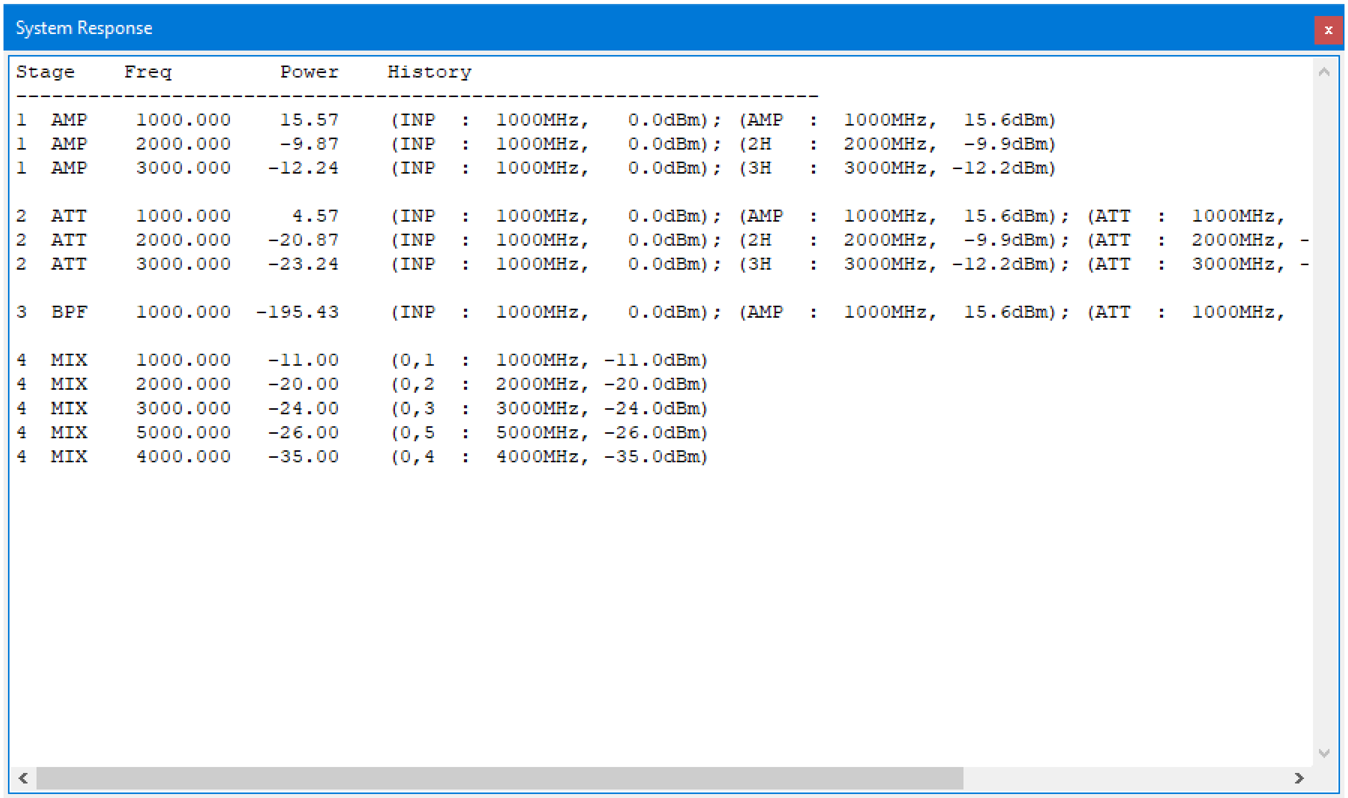


The RF test frequency (**Fin** in the **Input Signals** group) is assumed to be input to the system. To the right of each stage, a list of output frequencies, power levels, and signal histories display. On top of the line that connects stages, the desired signal propagation displays. Underneath the connecting line, the spurii or harmonics are listed.

The information for each output displays in a color-coded format. The information contains Freq, History, and Level but it may be shortened, depending on the Data Pattern selection. Click the **Change Data Pattern** button to experiment. Some typical formatted lines are decoded as follows:

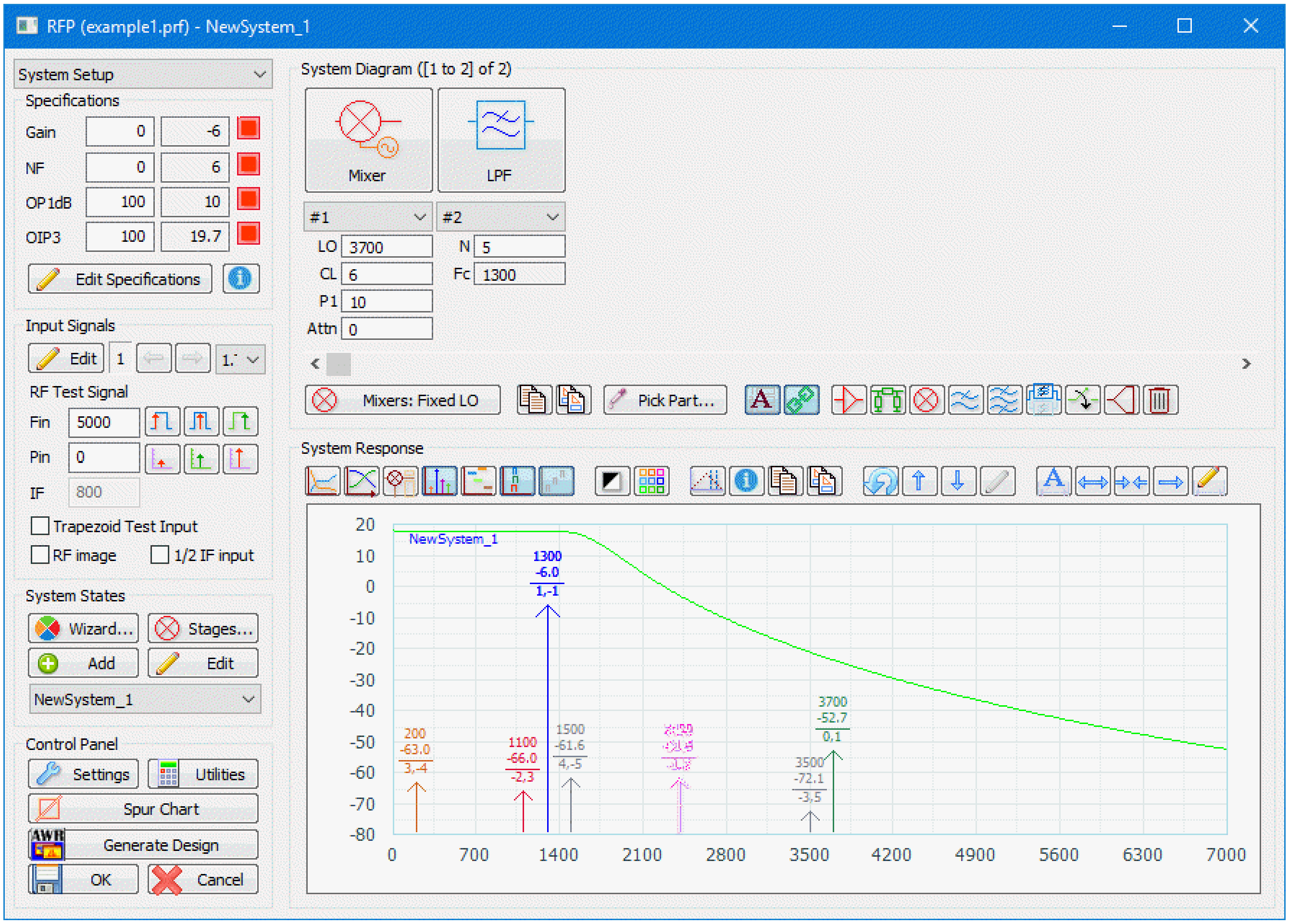
```
1000   0,1  -11   Fout=1000, Pout=-11dBm, it is calculated by IF=0*RF + 1*LO
2000  -1,3  -27   Fout=2000, Pout=-27dBm, it is calculated by IF=-1*RF + 3*LO
2000   2H   -57   Fout=2000, Pout=-57dBm, it is a 2nd harmonic.
```

To display all of the values stage-by-stage in a window, click the **i** (Information) button to display the System Response window.



13.15.6.5. Spot Freq Response View Mode

Spot Frequency Response is a view mode that displays the final output of the system for a spot frequency input. To select this mode, click the **Show Spot Freq Response** button on the System Response toolbar.



Each trace is color-coded with the same colors used in the Spur Tables. The signals are plotted in relation to Y-axis settings: Reference level and scale/div, which you can change with the **Toggle Left-axis scale/div** and **Increase/Decrease Left-axis Reference Level** buttons on the toolbar.

When a trace is resultant of a threat input, the arrow is red.

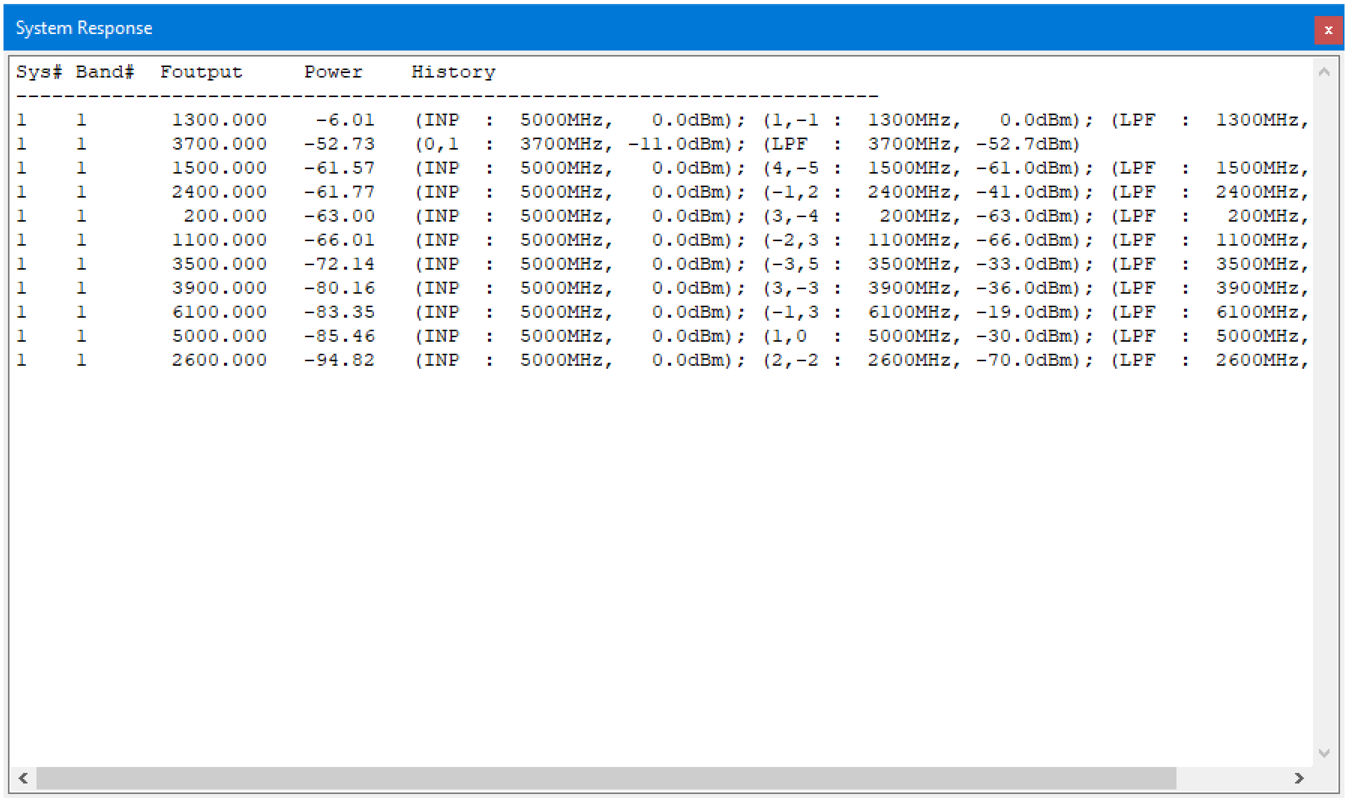
The RF test frequency (**Fin** in the **Input Signals** group) is assumed to be input to the system. The frequency shown on top of the traces corresponds to the calculated output frequencies.

On top of each trace, formatted data information displays. From top to bottom, the information reads Freq, Power, a separator line, and frequency history lines. Frequency history given as the bottom line corresponds to the first stage, and the top history line corresponds to the last stage. Frequency history contains data only for actual frequency changes. Components that do not contribute to frequency conversion (or harmonics, for example attenuators and filters) are not recorded as history.

If the system diagram contains a bandpass/lowpass filter through the end of stages, its response is plotted as an overlay. The filter trace draws in light green. Although the filter trace uses the scale/division of the graph, the reference level is ignored and the 0dB point of the trace is drawn near the top of the graph. The trace is provided for guidance only to understand how the output spectrum is shaped.

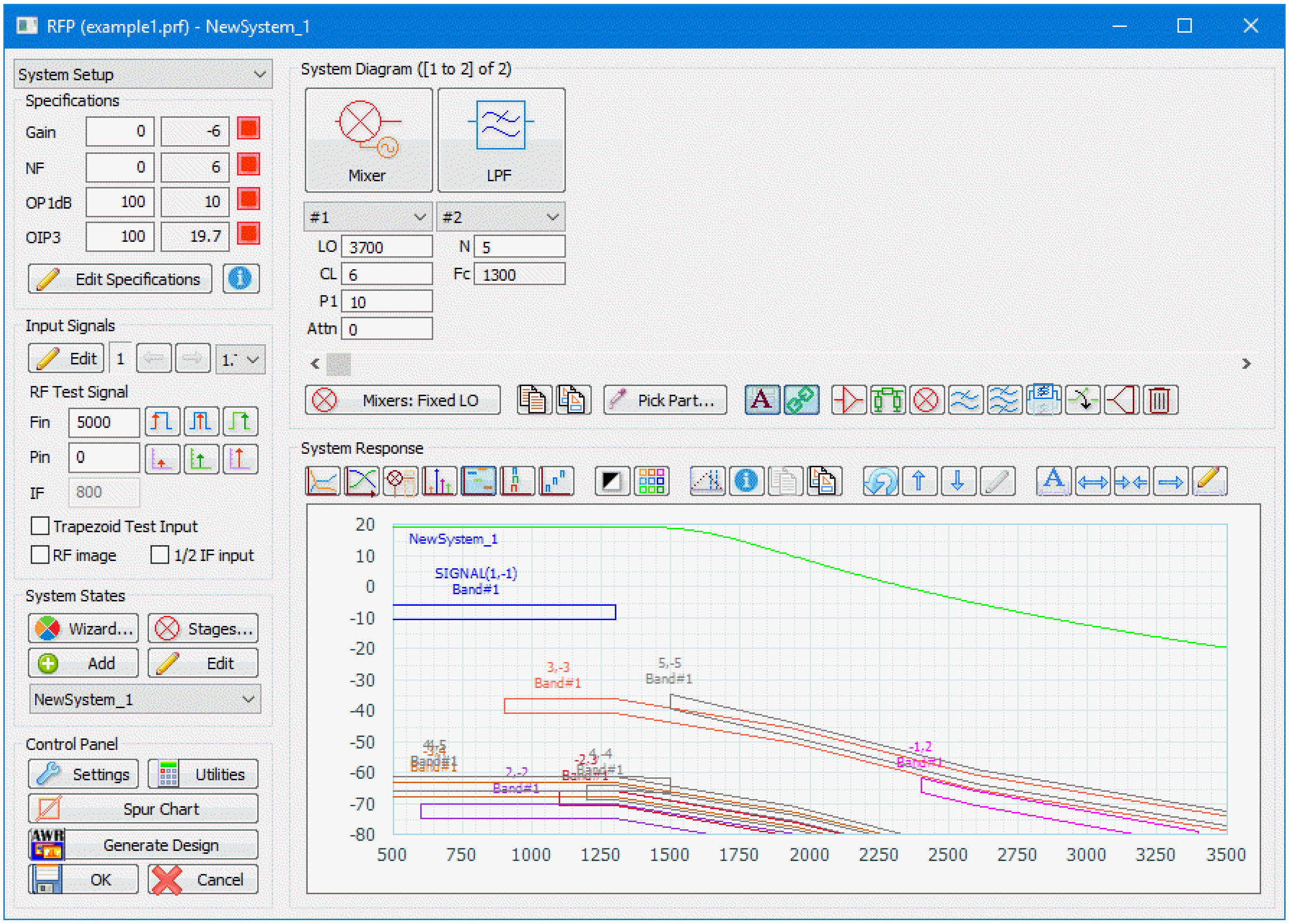
The filter trace is only drawn for the last filter in the diagram. The filter must not be followed by a mixer, either immediately, or after other components. Since the mixer converts the whole frequency spectrum, plotting a filter response for the spectrum before a mixer is pointless.

To display trace data in a window, click the **i** (Information) button to open the System Response window. Frequency information is also shown in the frequency history data.



13.15.6.6. Frequency Band Response View Mode

Frequency Band Response is a view mode that displays the final output of the system for a frequency band input. To select this mode, click the **Show Frequency Band Response** button on the **System Response** group toolbar.



Each trace is color-coded with the same colors used in the Spur Tables. The signals are plotted in relation to Y-axis settings: Reference level and scale/div, which you can change with the **Toggle Left-axis scale/div** and **Increase/Decrease Left-axis Reference Level** buttons on the toolbar.

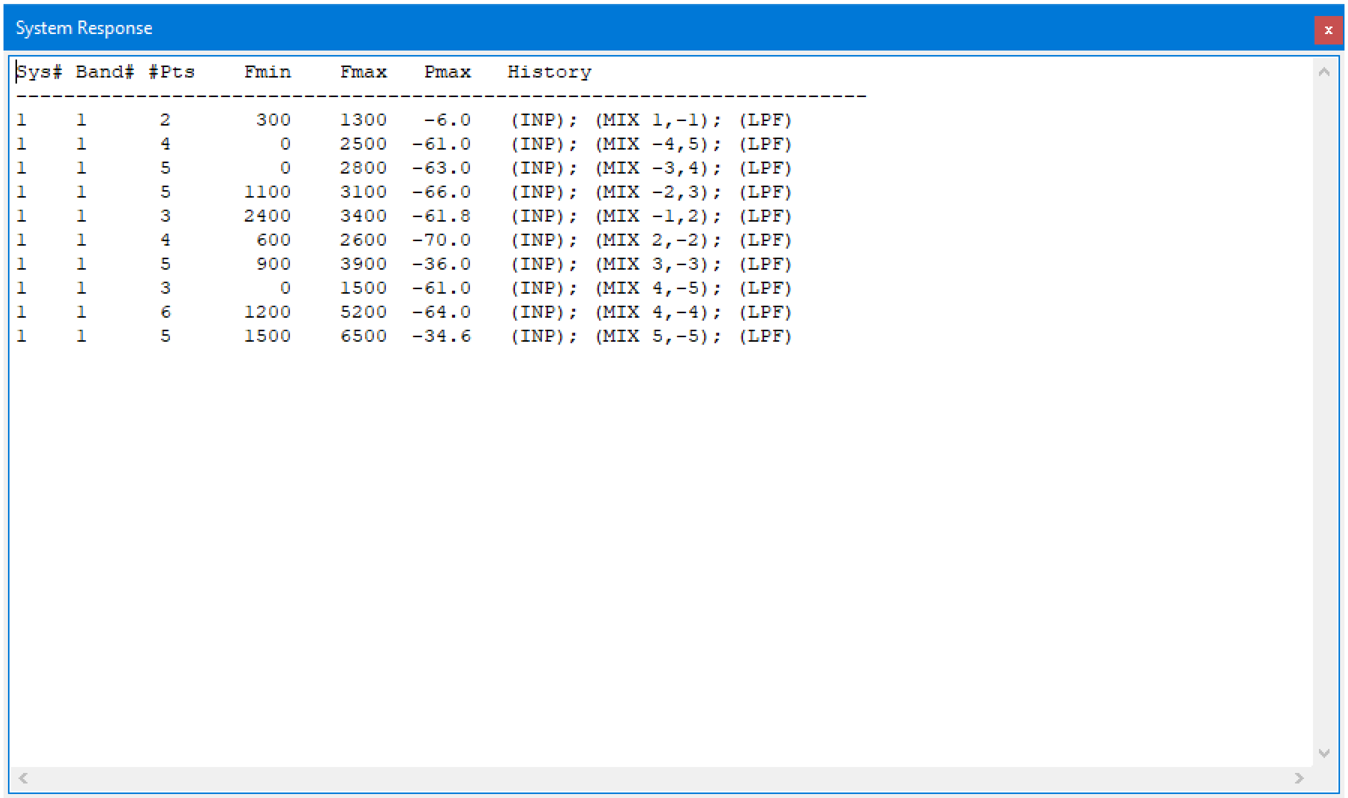
The RF test frequency band (as selected in the **Input Signals** group) is assumed to be input to the system. The band is processed through stages and it evolves into more than one band at the output of each stage. For example, each input band to an amplifier produces three outputs: fundamental, 2nd, and 3rd harmonic. The width of the input band may also become longer through stages. For example, a 1-2GHz input to an amplifier produces 1-2GHz fundamental, 2-4GHz 2nd harmonic, and 3-6GHz 3rd harmonic.

On top of each trace, formatted data information displays. From top to bottom, the information shows the frequency history lines. Frequency history given as the bottom line corresponds to the first stage, and the top history line corresponds to the last stage. Frequency history contains data only for actual frequency changes. Components that do not contribute to frequency conversion (or harmonics, for example attenuators and filters) are not recorded as history.

If the system diagram contains a bandpass/lowpass filter through the end of stages, its response is plotted as an overlay. The filter trace draws in light green. Although the filter trace uses the scale/division of the graph, the reference level is ignored and the 0dB point of the trace is drawn near the top of the graph. The trace is provided for guidance only, to understand how the output spectrum is shaped.

The filter trace is only drawn for the last filter in the diagram. The filter must not be followed by a mixer, either immediately, or after other components. Since the mixer converts the whole frequency spectrum, plotting a filter response for the spectrum before a mixer is pointless.

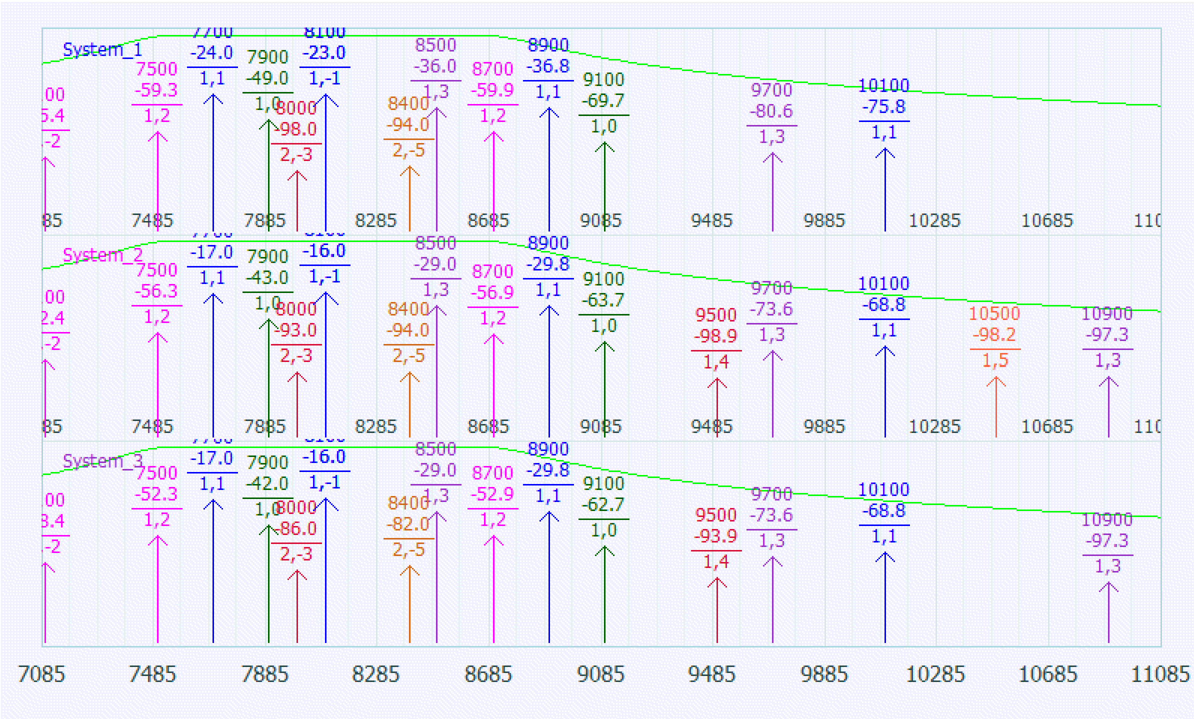
To display trace data in a window, click the **i** (Information) button to open the System Response window to view the number of points that make up the trace, its minimum and maximum frequency, and maximum power level.



Sys#	Band#	#Pts	Fmin	Fmax	Pmax	History
1	1	2	300	1300	-6.0	(INP); (MIX 1,-1); (LPF)
1	1	4	0	2500	-61.0	(INP); (MIX -4,5); (LPF)
1	1	5	0	2800	-63.0	(INP); (MIX -3,4); (LPF)
1	1	5	1100	3100	-66.0	(INP); (MIX -2,3); (LPF)
1	1	3	2400	3400	-61.8	(INP); (MIX -1,2); (LPF)
1	1	4	600	2600	-70.0	(INP); (MIX 2,-2); (LPF)
1	1	5	900	3900	-36.0	(INP); (MIX 3,-3); (LPF)
1	1	3	0	1500	-61.0	(INP); (MIX 4,-5); (LPF)
1	1	6	1200	5200	-64.0	(INP); (MIX 4,-4); (LPF)
1	1	5	1500	6500	-34.6	(INP); (MIX 5,-5); (LPF)

13.15.6.7. Viewing Responses of All Systems

To view the response of all systems in the Spot Freq Response or Frequency Response Band Response mode, click the **Plot All System States** button on the System Response toolbar.

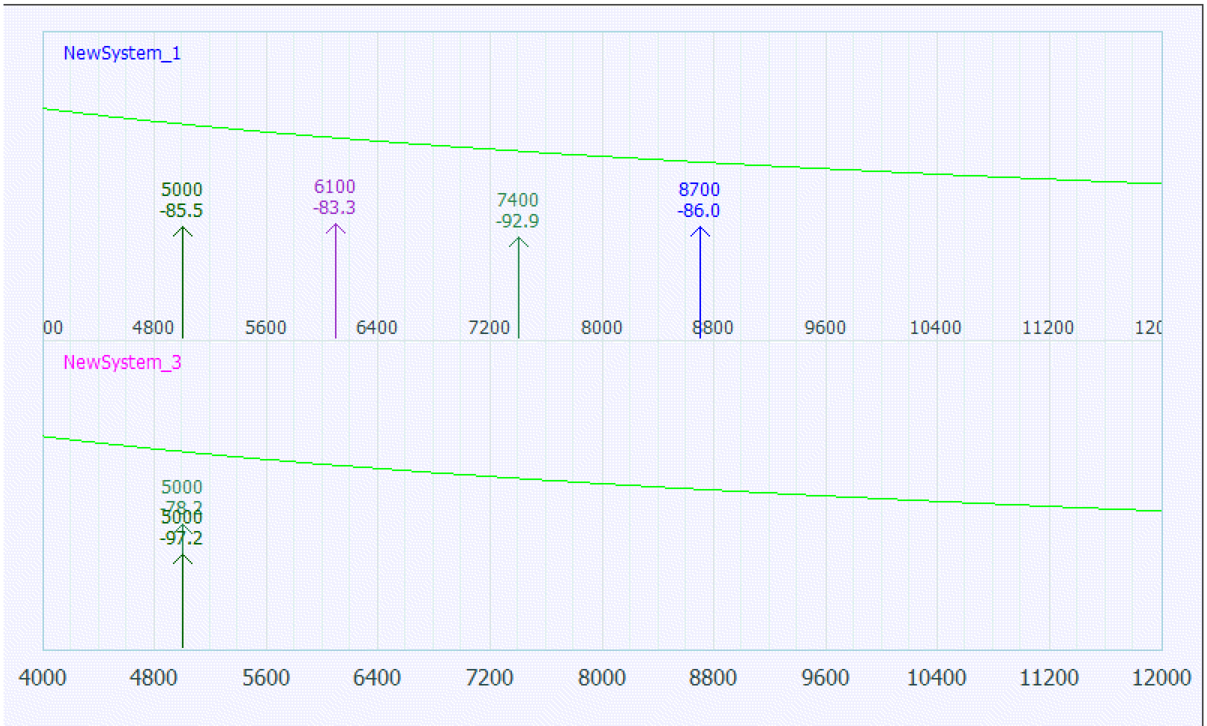


The response, spot frequency, or frequency band displays for all systems in the same plot. Depending on the number of systems in the design, the plot layout is automatically arranged in columns and rows.

```
Sys# Band#  Foutput      Power    History
--------------------------------------------------------------
1    1       5950.000    14.99    (INP  :  1000MHz,    0.0dBm); (BPF  :
1    1       5000.000    -0.09    (INP  :  1000MHz,    0.0dBm); (BPF  :
1    1       4950.000    -3.90    (0,1  :  6950MHz, -11.0dBm); (BPF  :
1    1       4950.000   -13.05    (INP  :  1000MHz,    0.0dBm); (BPF  :
1    1       6950.000   -33.97    (0,1  :  6950MHz, -11.0dBm); (BPF  :
1    1       6900.000   -47.56    (INP  :  1000MHz,    0.0dBm); (BPF  :
1    1       4050.000   -56.93    (INP  :  1000MHz,    0.0dBm); (BPF  :

2    1       5950.000    14.99    (INP  :  1000MHz,    0.0dBm); (BPF  :
2    1       5000.000    -0.09    (INP  :  1000MHz,    0.0dBm); (BPF  :
2    1       4950.000    -3.90    (0,1  :  6950MHz, -11.0dBm); (BPF  :
2    1       4950.000   -13.05    (INP  :  1000MHz,    0.0dBm); (BPF  :
2    1       6950.000   -33.97    (0,1  :  6950MHz, -11.0dBm); (BPF  :
2    1       6900.000   -47.56    (INP  :  1000MHz,    0.0dBm); (BPF  :
```

13.15.6.8. Viewing Spot/Band Responses of All Stages

To view the response of conversion stages of the selected system in the Spot Freq Response or Frequency Band Response mode, click the **Plot Conversion Stages** button on the **System Response** group toolbar.



The response, spot frequency, or frequency band displays for all conversion stages in the same plot. The input signal, spot, or band, is shown as stage 0. The first conversion stage (the first mixer output), displays as stage 1, and so on.

For every stage, a filtering trace can display if there is such a filter following a mixer output.

13.15.7. Generating Designs in the AWR Design Environment Software

After initial frequency planning is finished in the RFP RF Planning Tool Wizard, you can generate the design in the AWR Design Environment software as an VSS project for further detailed analysis and optimization. The Generate Design dialog box allows you to generate a VSS system diagram for the selected system in the RFP. To access this dialog box, click the **Generate Design** button in the main RFP dialog box.

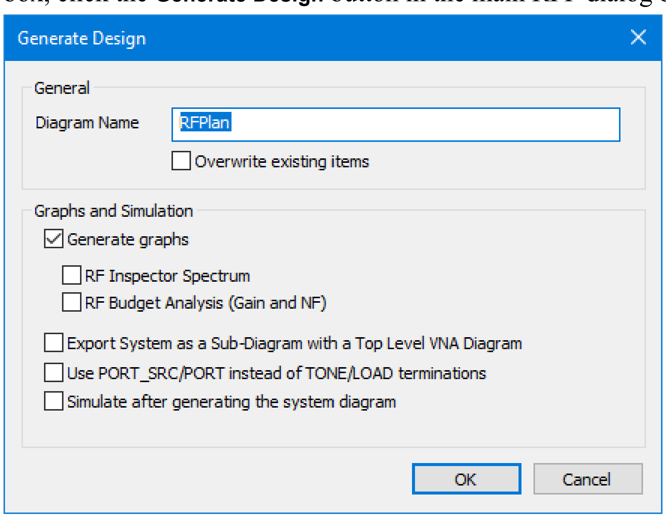


Diagram Name is the title of the VSS diagram to generate.

Select the **Overwrite existing items** check box to overwrite generated items in the AWR Design Environment software. If the generated item already exists and this check box is not selected, RFP does not create the new item and a warning that the item already exists is issued.

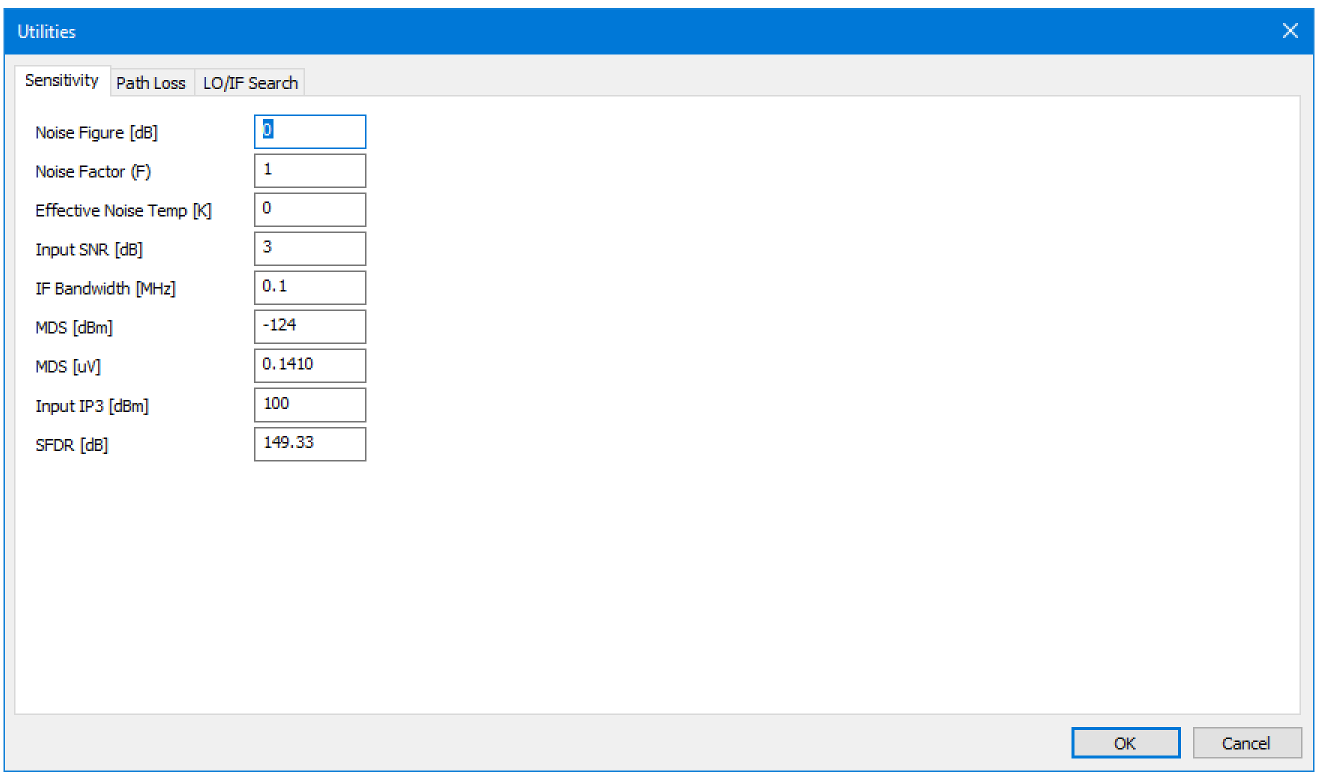
In the **Graphs and Simulation** group, there are options to generate analysis setups and initiate them upon generation. Click **Generate graphs** and one or both of the RF check boxes beneath it to create graphs. Select the **Simulate after generating the system diagram** check box to start analysis after generation.

13.15.8. Utilities

RFP provides some popular utilities for system design. To access the following utilities, click the **Utilities** or **Spur Chart** buttons in the main RFP dialog box to display the Utilities dialog box or Spur Chart dialog box respectively.

13.15.8.1. Sensitivity

The Sensitivity utility provides a quick way to calculate system sensitivity and related information. To access the Sensitivity utility, click the **Utilities** button in the main RFP dialog box to display the Utilities dialog box, then click the **Sensitivity** tab.



Sensitivity or minimum discernible signal (MDS) [dBm] of a receiver system is defined by the following equation:

```
MDS = -174 + 10*log10 (IFBW) + NF + SNR
```

where `IFBW` is the IF bandwidth [Hz], `NF` is the input referred noise figure [dB] and `SNR` is the minimum required signal to noise ratio for reception of a signal [dB].

Spurious free dynamic range (**SFDR**) of a system is given by the difference between maximum and minimum receivable signal levels:

```
SFDR = (MDS + 2*IIP3)/3 - MDS
```

where `IIP3` is the third-order input intercept point.

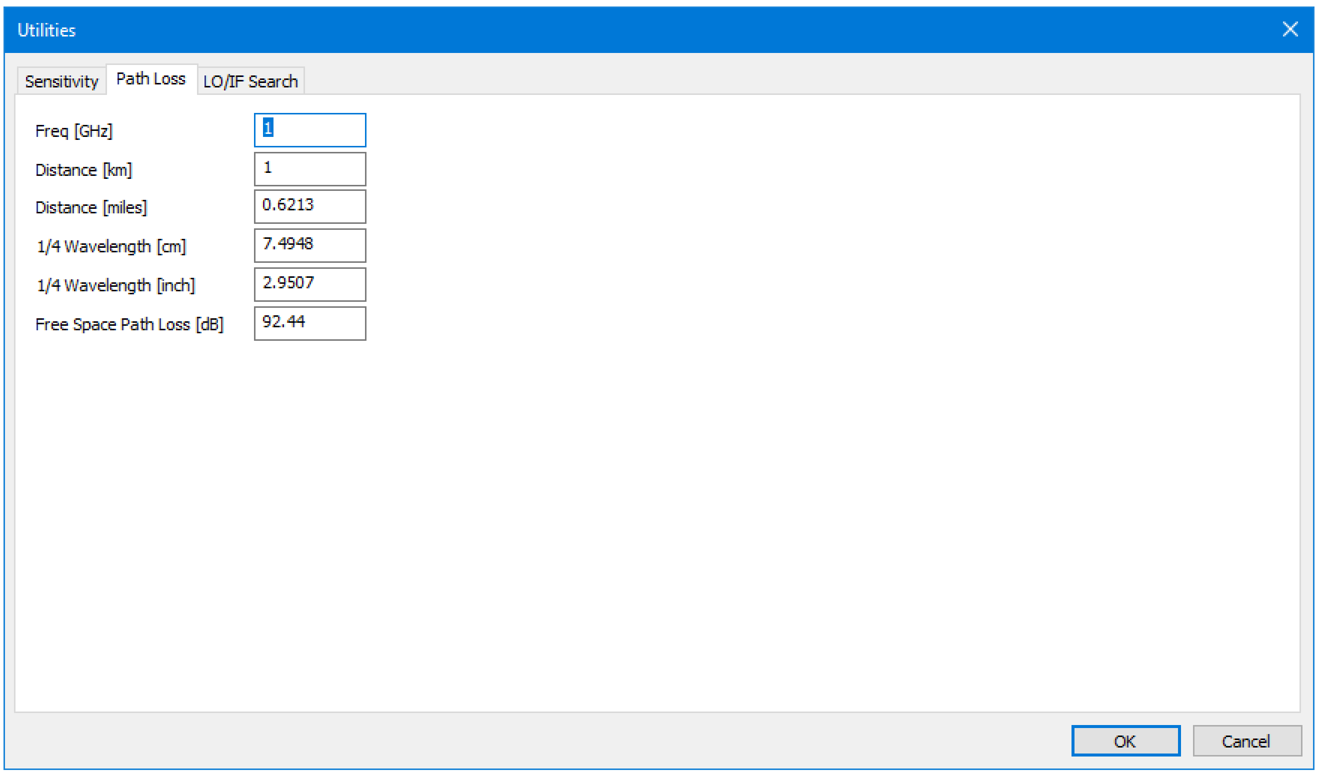
MDS and **SFDR** are the resultant values and therefore grayed. All other parameters are available for editing. **Noise Figure**, **Noise Factor (F)** and **Effective Noise Temp [K]** are interrelated and you can specify any of them:

```
NF = 10 * log10 (F) Te = 290 * (F-1)
```

MDS[uV] is the microvolts equivalent of **MDS[dBm]** of power on 50 ohms load.

13.15.8.2. Path Loss

The Path Loss utility provides a quick way to calculate free space path loss. To access the Path Loss utility, click the **Utilities** button in the main RFP dialog box to display the Utilities dialog box, then click the **Path Loss** tab.



Free Space Path Loss is given by:

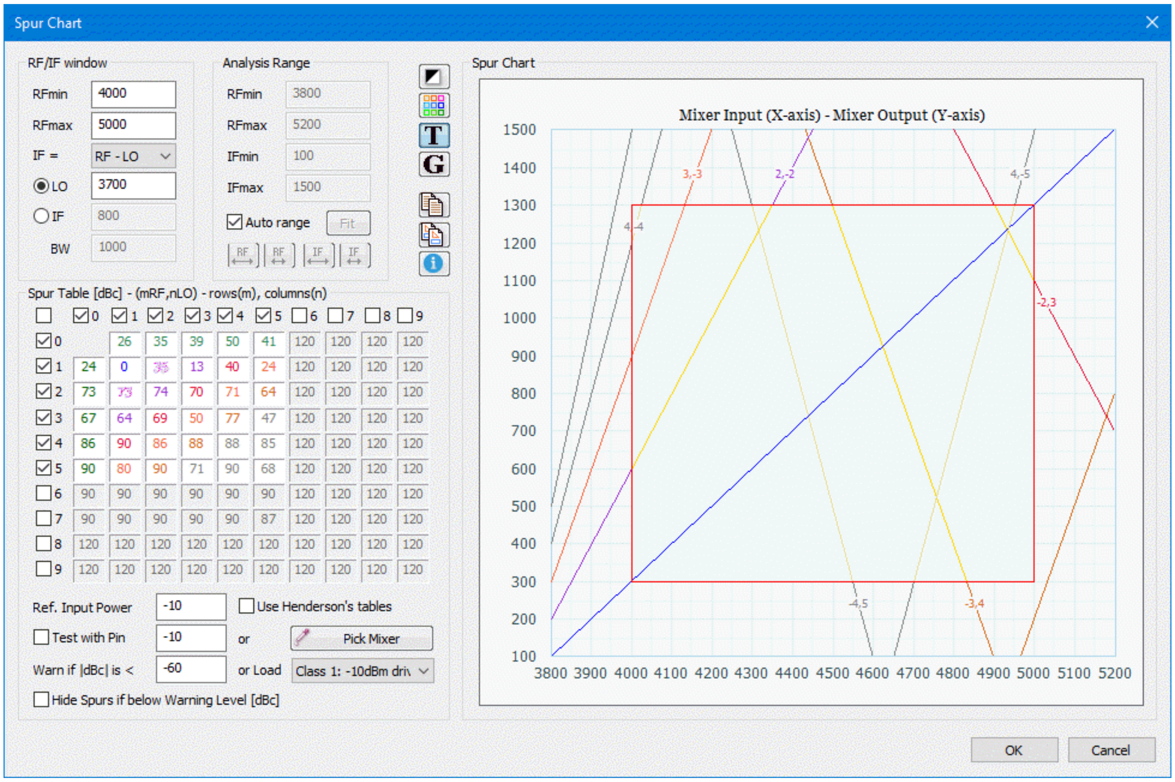
```
Path Loss = 92.44 + 20*log10(D) + 20*log10(F)
```

where D is the distance from the transmitter and receiver [km] and F is the frequency [GHz].

In this dialog box you specify **Freq** (frequency) and **Distance**, while wavelength and **Free Space Path Loss** are resultant values, so uneditable.

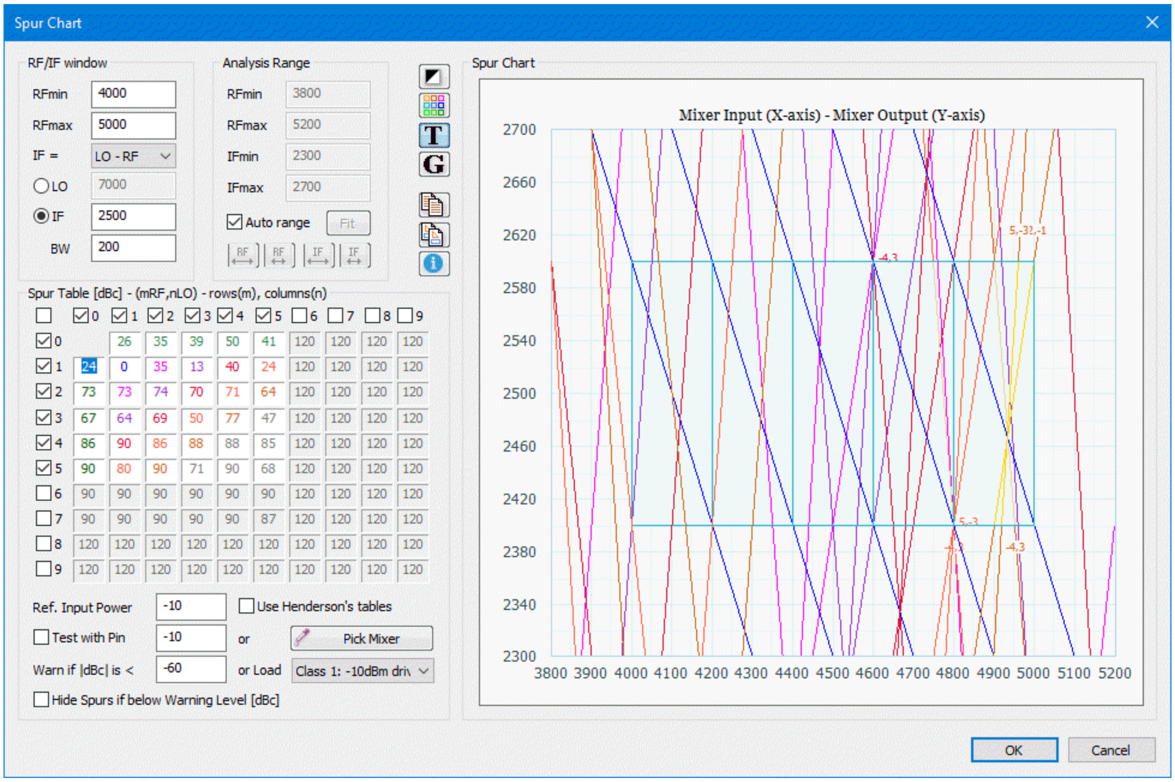
13.15.9. Spur Chart

Spur Chart is a conventional and useful technique to assess spurious free regions for a single-stage frequency conversion. To access spur chart, click the **Spur Chart** button under **Control Panel** in the main dialog box.



In the **RF/IF Window** group, you enter the main frequency conversion parameters. **RFmin** and **RFmax** determine the range for the input frequencies. You specify a conversion scheme in the **IF** drop-down. For up-conversion select **RF+LO** and for downconversion, select **RF-LO** or **LO-RF**. Next, select the behavior of **LO**:

- For fixed LO conversions, select **LO**. RFP tries to map the center of the desired RF band into the center of the IF band by using the specified LO. For example, if RF=4000-5000, and LO=7000, and **LO-RF** is selected, IF is centered at 7000-4500=2500. In the previous figure, RF is plotted on the X-axis and IF is plotted on the Y-axis with the band centers clearly seen. The RF-to-IF mapping window displays in light blue.
- For variable LO conversion, select **IF** and specify **IF** for the target IF center and **BW** for the sub-band widths. RFP calculates the necessary LO values to map separate RF sub-bands into the same IF window. For example, the following figure shows 200MHz sub-bands being converted into IF=2500. The LO value for the sub-band 4000-4200 is 6600, and for the 4200-4400 it is 6800, and so on. The plot shows five separate IF windows, with spurii coming from all conversions plotted in the same place. It is useful to plot multiple spurii when RF channelizing filter is not used. In this case, spurii are generated for all signals in the RF band, not only the sub-band of interest. This conversion mode makes it easy to visualize all different LO conditions in one plot.



The **Analysis Range** group is used to determine the X- and Y- ranges for the chart. If auto-scaling is needed, select the **Auto range** check box. To quickly fit and center the IF window in the chart, click the **Fit** button.

The **Spur Table** group presents the spurii levels in clickable rows and columns for mxn products. You display the desired m or n products by selecting the row or column. Rows correspond to m and columns correspond to n in +/-mRF+/-nLO.

Table entries and plotted traces are color-coded. If you click on a table entry, the corresponding trace is highlighted on the plot. Similarly, clicking on a trace highlights the corresponding table entry.

You specify the Spur Table reference input power in **Ref. Input Power**. The entries are automatically populated if a predefined Mixer class is specified by changing the **Load**. This selection is only used to load values into the Spur Table and is not used afterwards. You can also populate the table by clicking the **Pick Mixer** button to select a mixer from the part library.

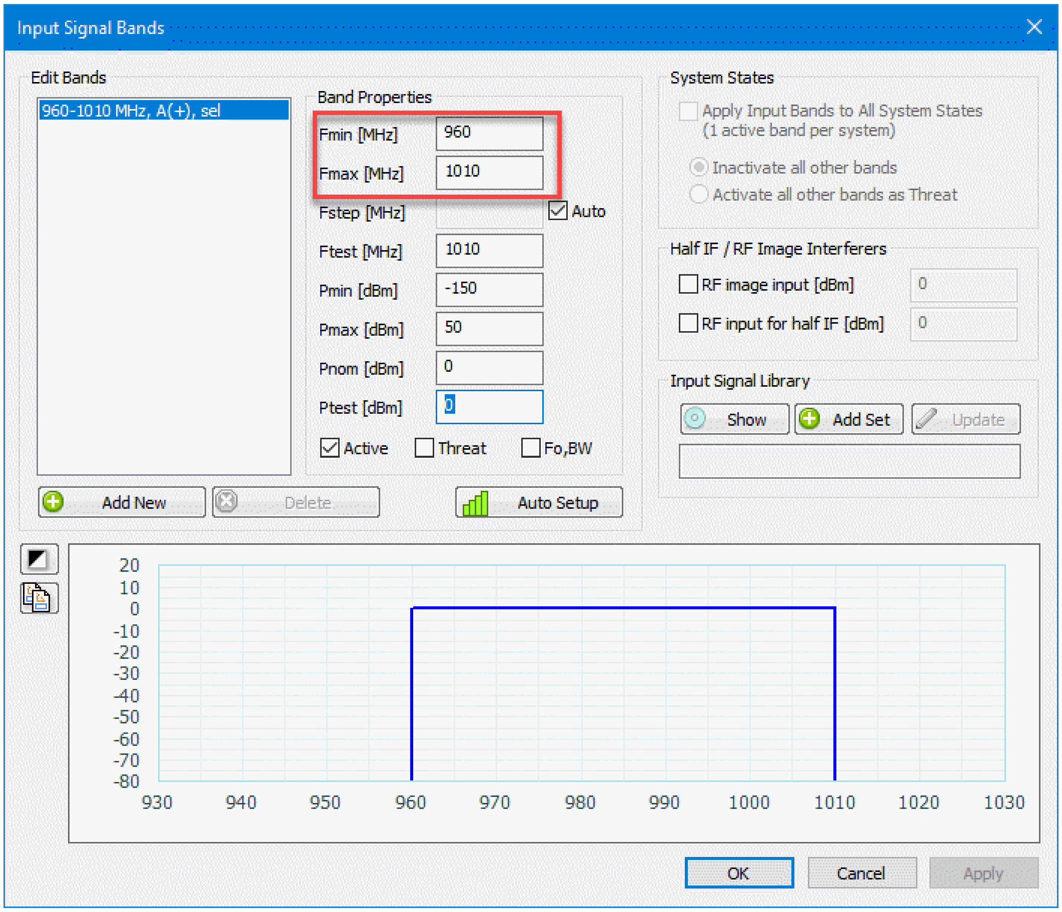
You can display the actual spurii levels by selecting the **Test with Pin** check box to calculate the actual spurii levels in [dBm] and display them in the grayed-out table. The entries are not editable when this check box is selected.

Often, many less important spurii traces clutter the Spur chart. You can hide from the plot those that fall below a set threshold by selecting **Hide Spurs if below Warning Level [dBc]** and specifying the threshold in **Warn if |dBc| is <**.

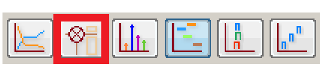
13.15.10. RFP RF Planning Tool Wizard Example

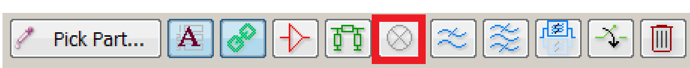
The following example demonstrates RFP RF Planning Tool Wizard capabilities. A final step for VSS program use is also presented. Any changes you make prompt the wizard to recalculate the auto parameters, analyze the system response, and display all results in graphs and windows. No action is necessary to update the results as they are live data.

1. In the main RFP dialog box, under **System States**, click the **Wizard** button to display the Select Wizard Action dialog box.
2. Select **Clear Systems and Input Bands**.
3. In the main RFP dialog box, under the **Input Signals** group, click the **Edit** button to display the Input Signal Bands dialog box.
4. Under **Band Properties**, change **Fmin** to "**960**" and **Fmax** to "**1010**", then click **OK**.

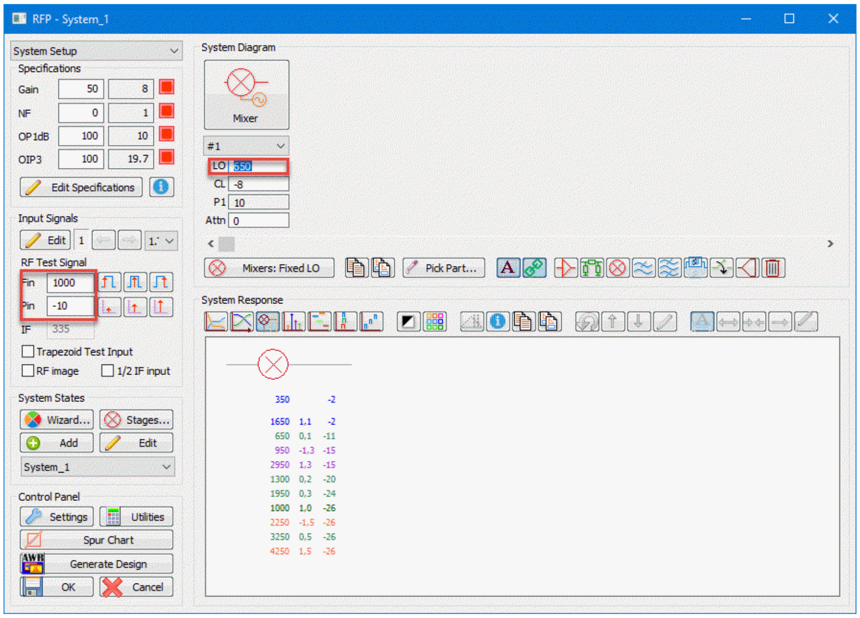


5. In the main RFP dialog box, on the System Response toolbar, click the **Show Spot Freq Schematic** button, then click the **Add Mixer** button at the bottom of the System Diagram window, as shown in the following figure.

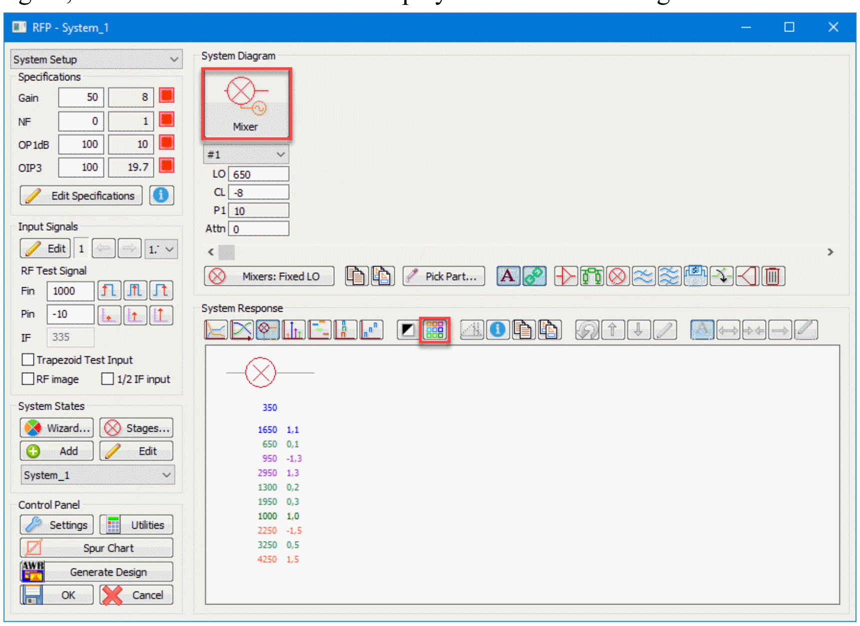




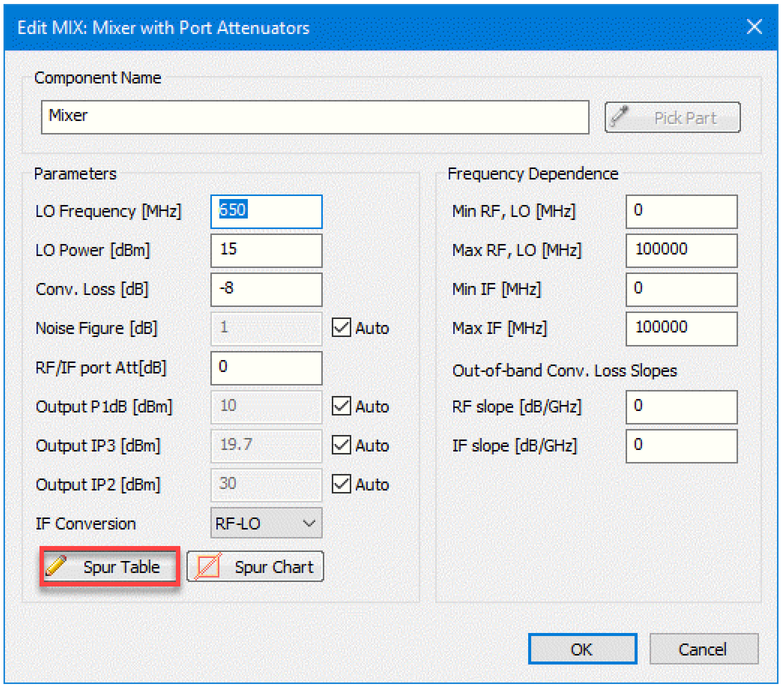
6. Change **LO** to "**650**" (MHz), and under **RF Test Signal**, set **Fin** to "**1000**" (MHz) and **Pin** to "**-10**" (dBm).



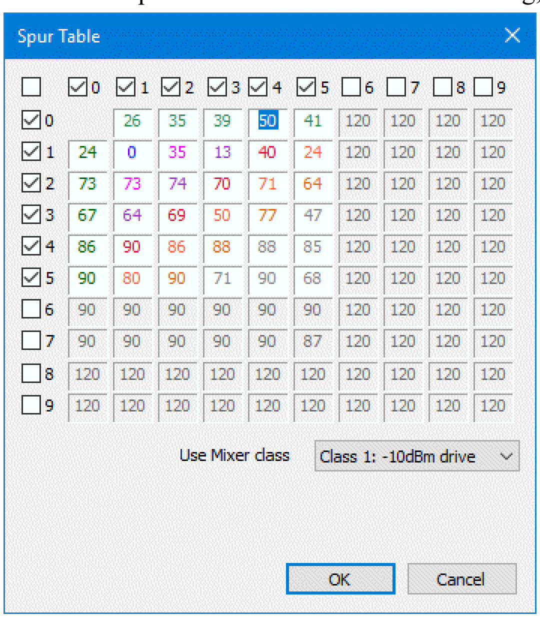
7. Click the **Change Data Pattern** button to change the display of IM products and power levels, as shown in the following figure, then click the **Mixer** icon to display the Edit MIX dialog box.



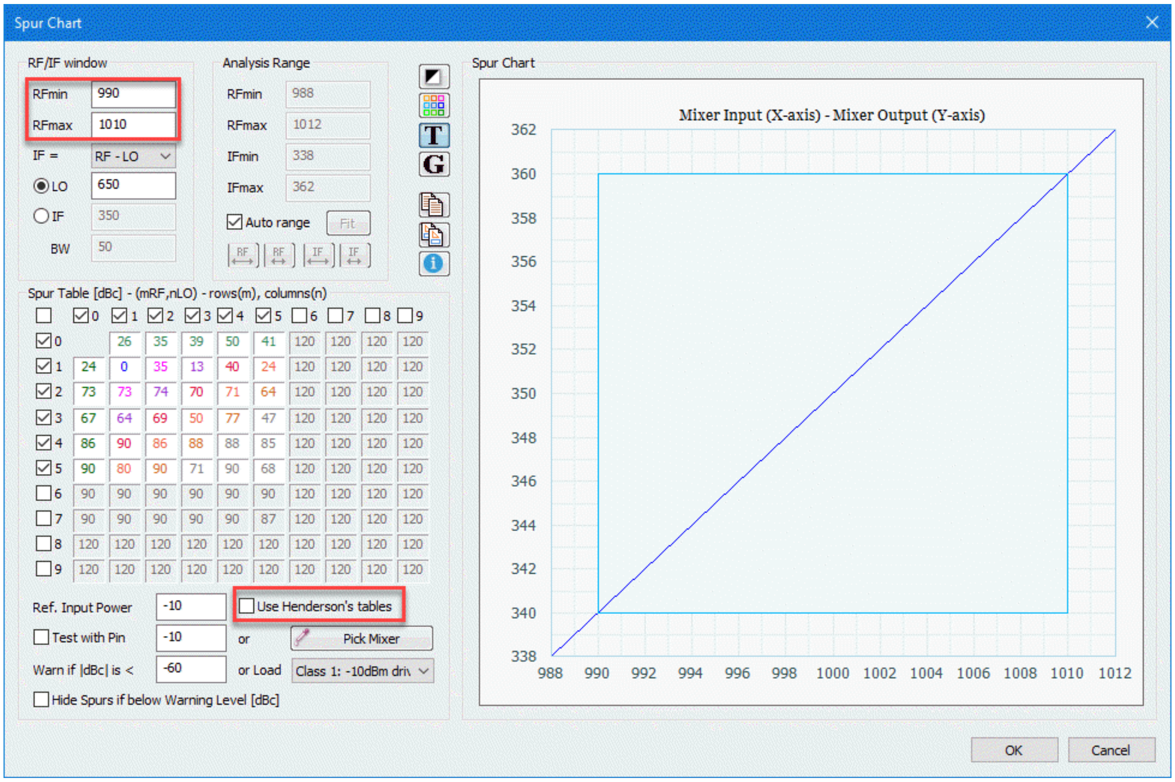
8. In the Edit MIX dialog box, click the **Spur Table** button to display the Spur Table dialog box.



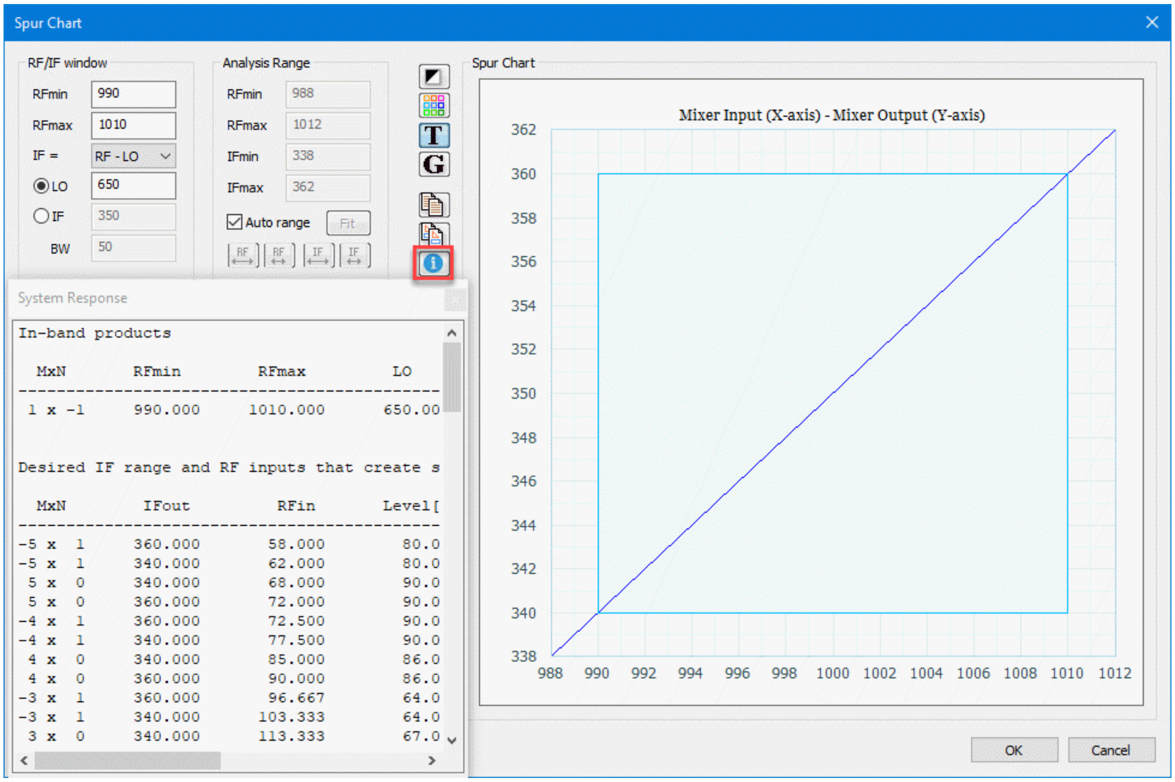
9. Ensure the Spur Table values match the following, then click **OK**.



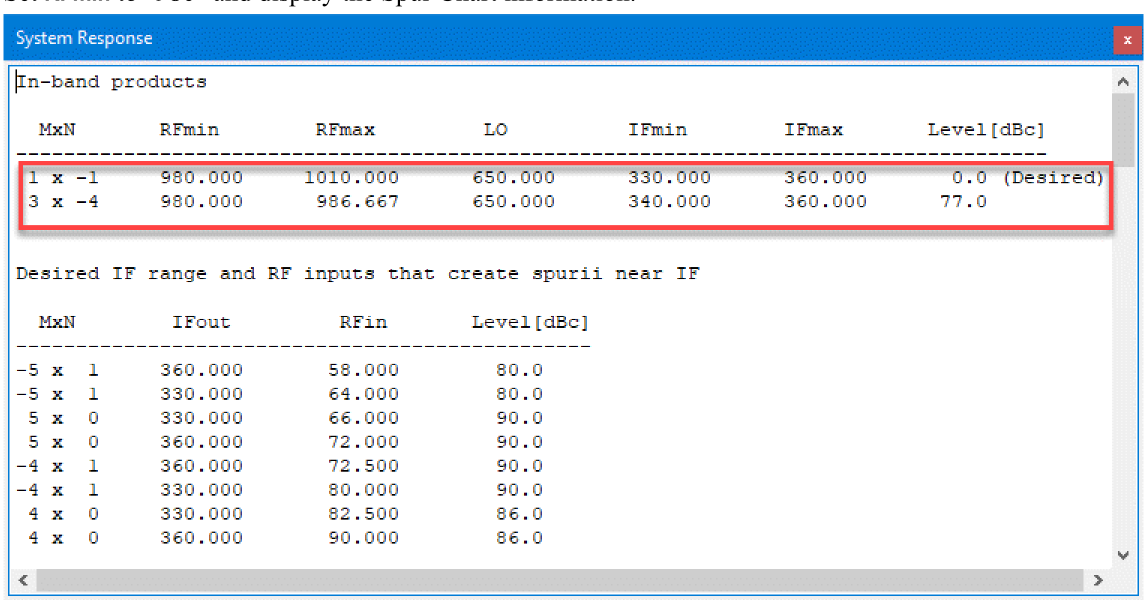
10. Click the **Spur Chart** button next to the **Spur Table** button to display the Spur Chart dialog box.
11. Under **RF/IF window**, set **RFmin** to "**990**" (MHz) and **RFMax** to "**1010**" (MHz). You can also optionally select **Use Henderson's tables** to use a pre-defined behavioral mixer.



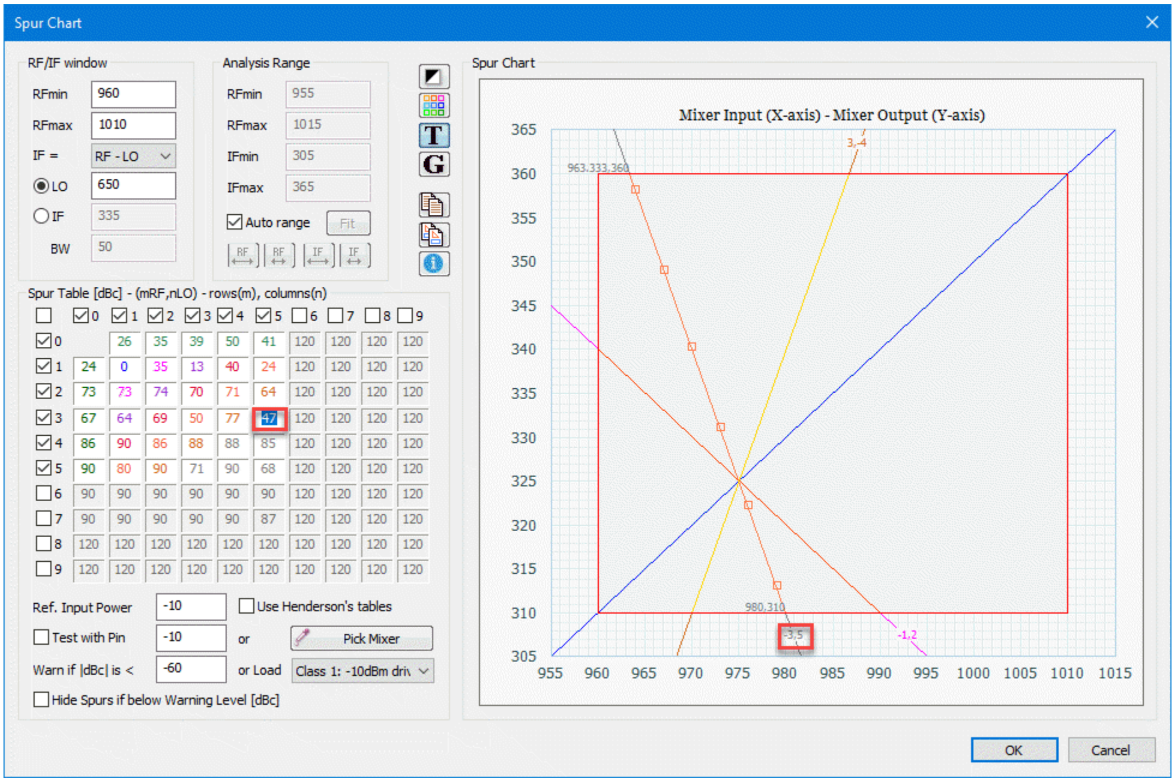
12. Click the **Show/Hide Plot Info** button to toggle the display of Spur Chart information in tabular format in a System Response window. You can leave this window open while changing any frequency in the Spur Chart by selecting the frequency and typing a new value or scrolling the mouse wheel to change the frequency in 10 MHz steps (or press the **Ctrl** key while scrolling to change the frequency in 1 MHz steps). When you change a frequency, the Spur Chart and System Response windows both reflect the change.



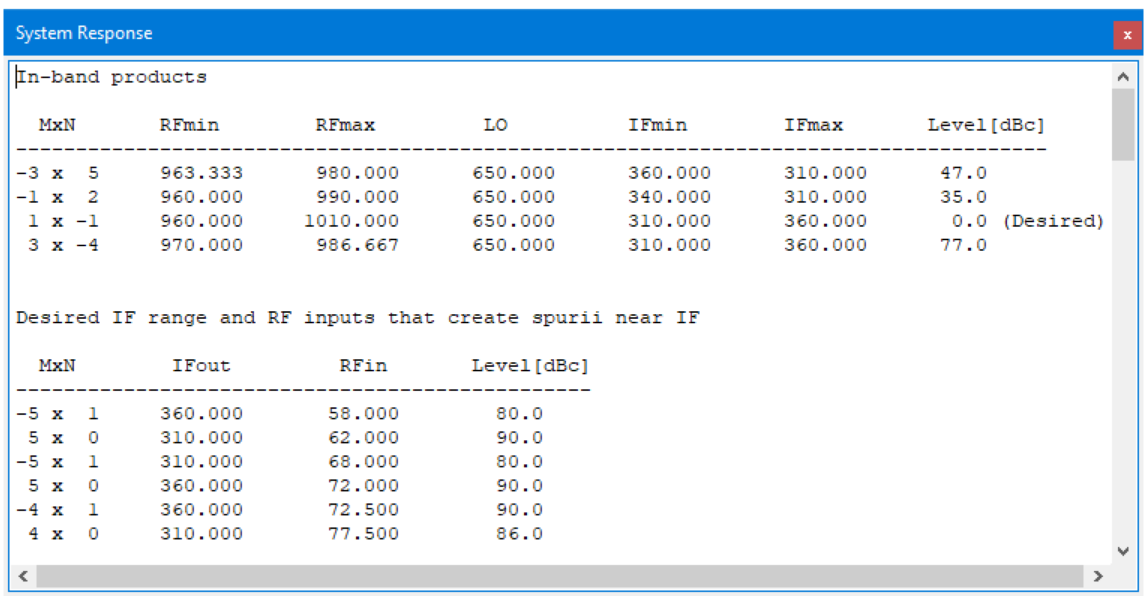
13. Set **RFmin** to "**980**" and display the Spur Chart information.



14. Click the **Show/Hide Plot** button to hide the Spur Chart information, then set **RFmin** to "**960**" and click outside of the option. Three more spurs are generated in the band and shown as new lines in the Spur chart.
15. Click on trace (-3,5) and note that the dBc value is highlighted in the Spur Table.



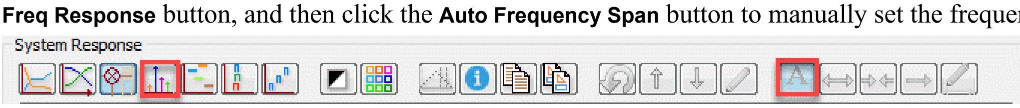
16. Click the **Show/Hide Plot** button to display the Spur Chart information. The band spurs shown in tabular format show **RFMin** as 960 MHz and **RFmax** as 1010 MHz.



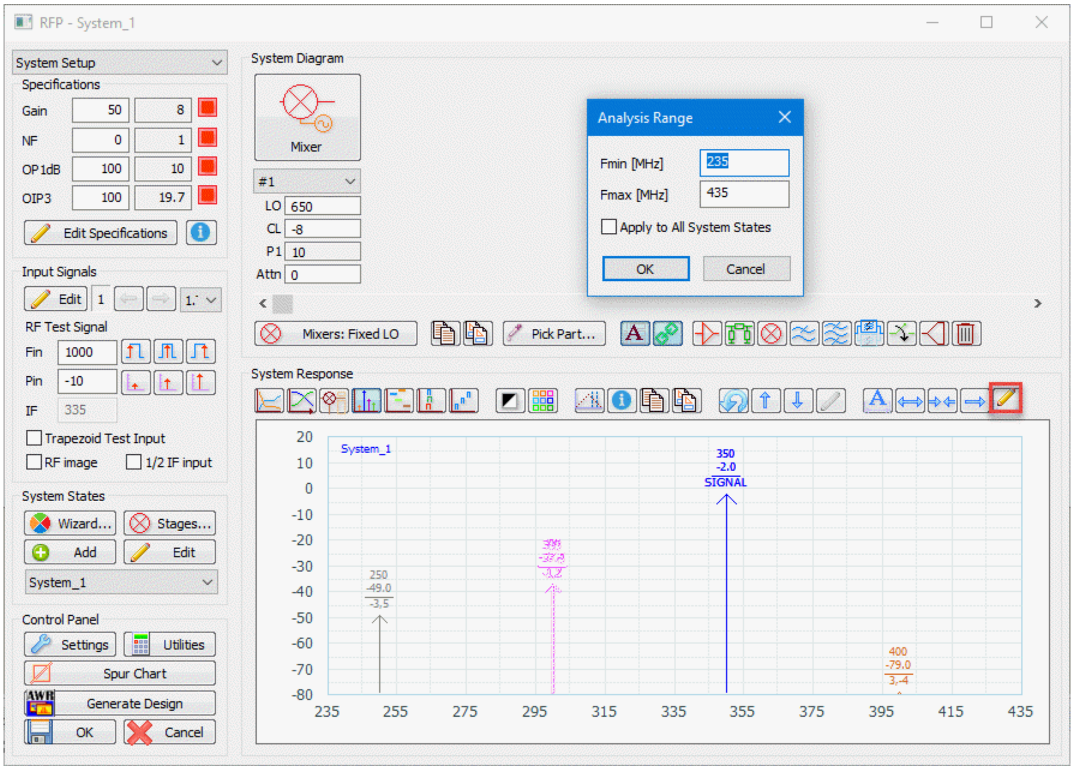
System Response

```
In-band products

  MxN        RFmin        RFmax        LO          IFmin        IFmax        Level[dBc]
---------------------------------------------------------------------------------------
-3 x  5     963.333      980.000      650.000      360.000      310.000      47.0
-1 x  2     960.000      990.000      650.000      340.000      310.000      35.0
 1 x -1     960.000     1010.000      650.000      310.000      360.000       0.0 (Desired)
 3 x -4     970.000      986.667      650.000      310.000      360.000      77.0


Desired IF range and RF inputs that create spurii near IF

  MxN         IFout         RFin        Level[dBc]
------------------------------------------------
-5 x  1     360.000       58.000        80.0
 5 x  0     310.000       62.000        90.0
-5 x  1     310.000       68.000        80.0
 5 x  0     360.000       72.000        90.0
-4 x  1     360.000       72.500        90.0
 4 x  0     310.000       77.500        86.0
```

17. Close all open windows and return to the main RFP dialog box. On the System Response toolbar, click the **Show Spot Freq Response** button, and then click the **Auto Frequency Span** button to manually set the frequency span.

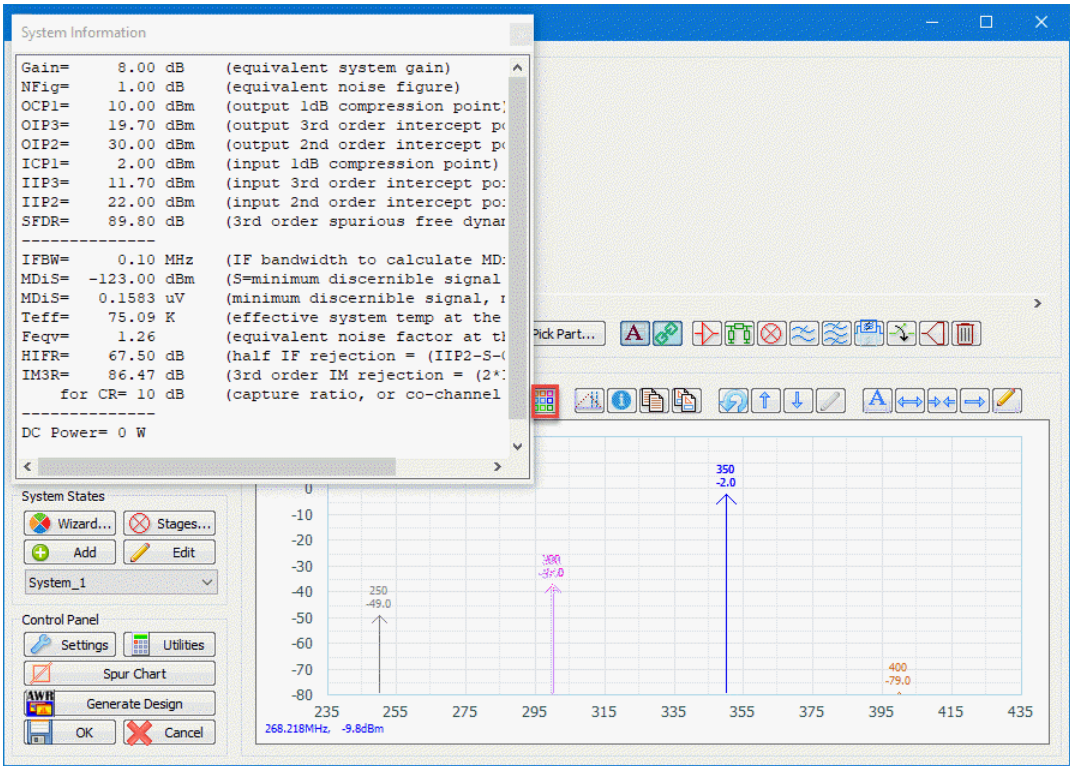


18. Click the **Set Frequency Span** button to display the Analysis Range dialog box. Set the **Fmin** and **Fmax** values as shown, then click **OK**.

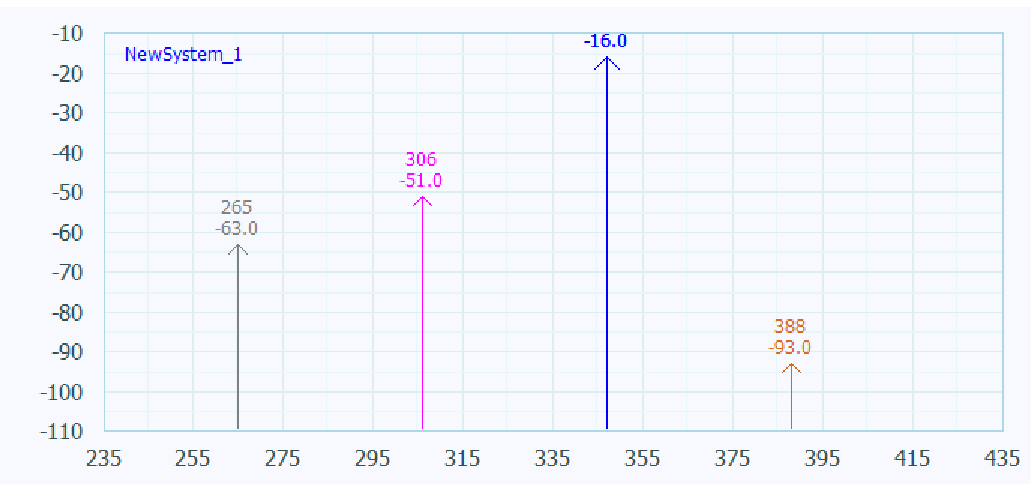


19. You can use the **Increase Left-axis Reference Level** and **Decrease Left-axis Reference Level** toolbar buttons to adjust the left axis scale, and the **Toggle Left-axis scale/div** button to adjust y-axis scaling 1dB/div, 2dB/div, 5dB/div and 10dB/div.

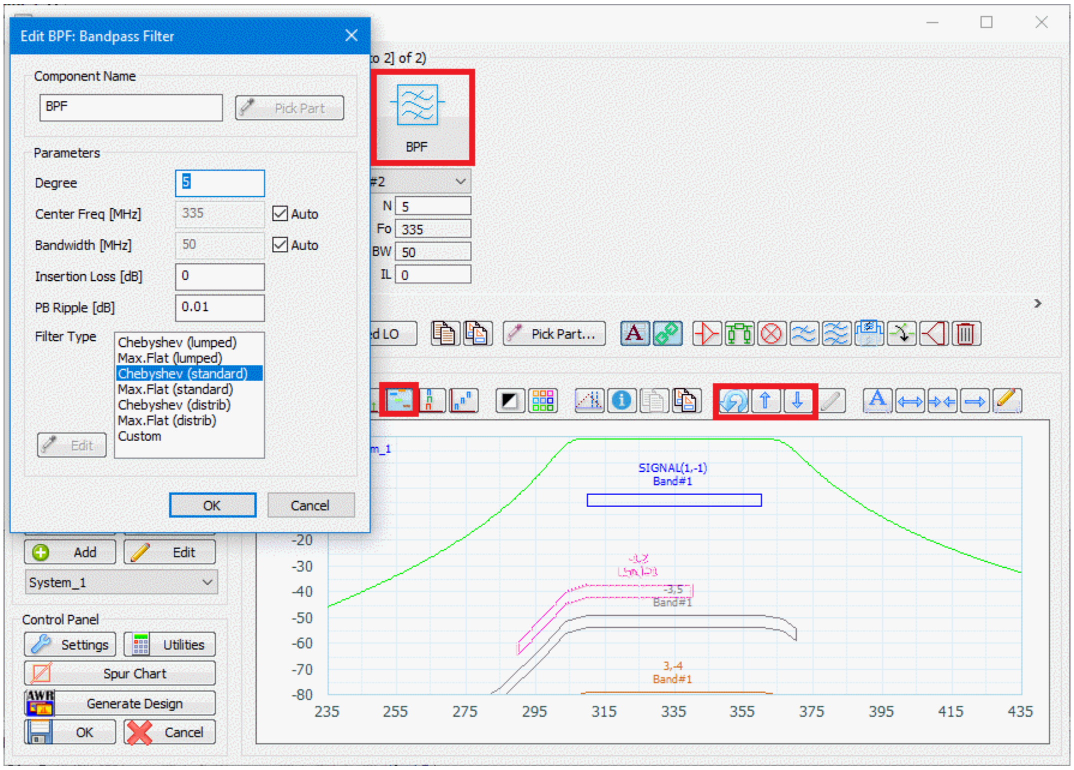
20. The spectrum plot displays as follows. You can click the **Change Data Pattern** button to change the values that display on the spurs.



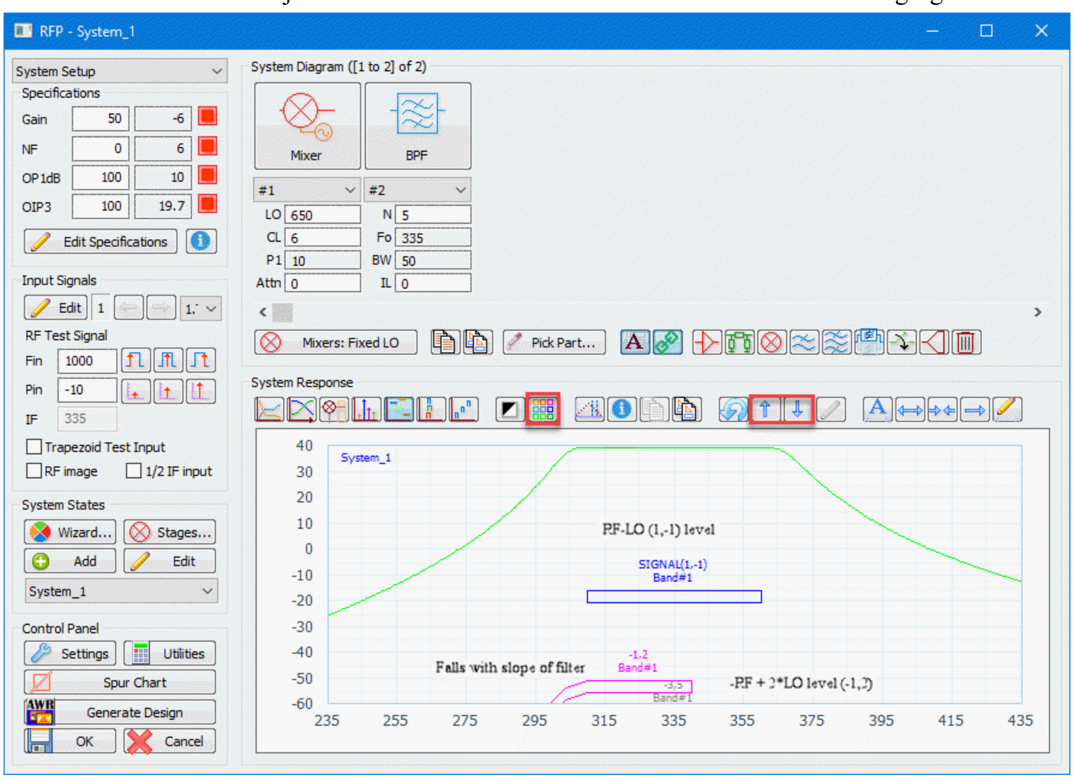
21. Select the **LO** frequency and press the **Ctrl** key while scrolling the mouse wheel to change its value in 1 MHz steps. As you sweep, the LO the spectrum changes. The following figure shows **LO** set to 653 MHz. After viewing the results, set **LO** back to "**650**" MHz.



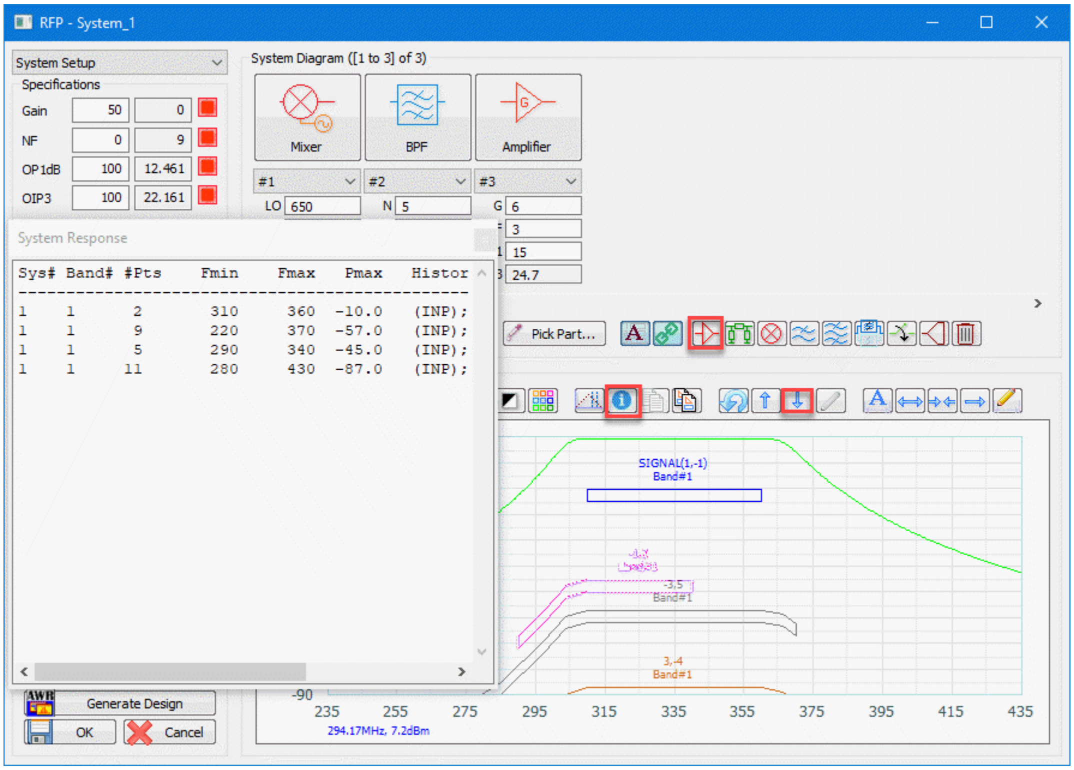
22. Use the **Increase Left-axis Reference Level** and **Decrease Left-axis Reference Level** toolbar buttons to adjust the left axis scale, and the **Toggle Left-axis scale/div** button to adjust y-axis scaling to the values shown in the following figure. Click the **Add Bandpass Filter** button to add a filter, and then click the BPF icon to display the Edit BPF: Bandpass Filter dialog box. Note the settings in this dialog box and then close it. Click the **Show Frequency Band Response** button and note the filter response displayed on the graph.



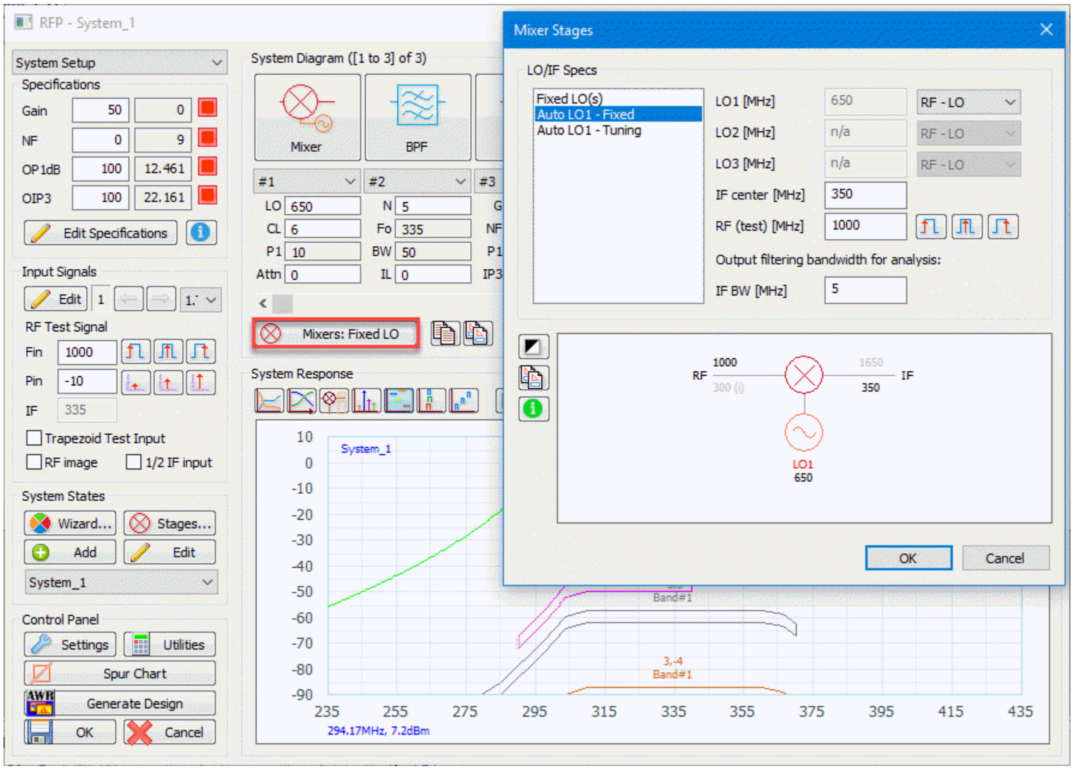
23. Click the **Change Data Pattern** button and use the **Increase Left-axis Reference Level** and **Decrease Left-axis Reference Level** toolbar buttons to adjust the left axis scale to the values shown in the following figure.



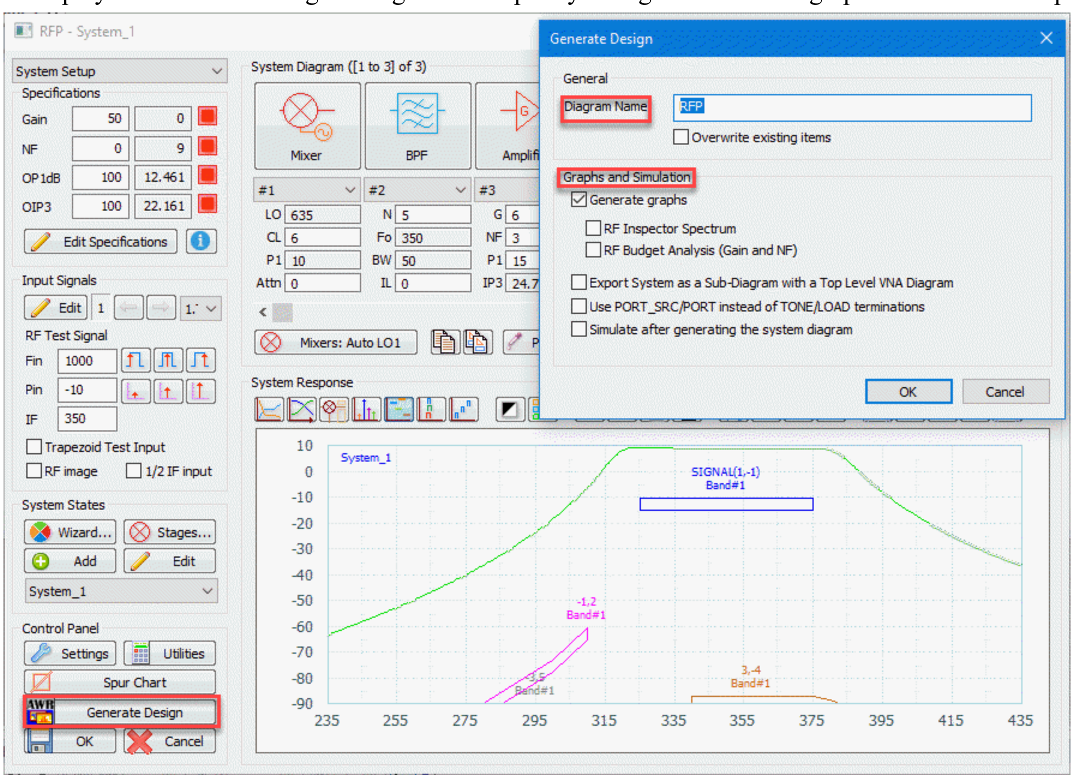
24. Click the **Add Amplifier** button to add an amplifier, and then on the System Response toolbar, click the **Show Plot Information** button to display the System Response window with Spur Chart information in tabular format. Click the **Decrease Left-axis Reference Level** button to adjust the left axis as shown in the following figure.



25. Click the **Mixers: Fixed LO** button to display the Mixer Stages dialog box. In the Mixer Stages dialog box, select **Auto LO1 - Fixed** to fix the IF. Now as you sweep on the RF the LO tracks it.



26. To export the design to the AWR Design Environment platform, under **Control Panel**, click the **Generate Design** button to display the Generate Design dialog box and specify a diagram name and graph and simulation options.



NOTE: Subsequent new designs use the latest frequency plan properties instead of requiring you to re-enter specifications.

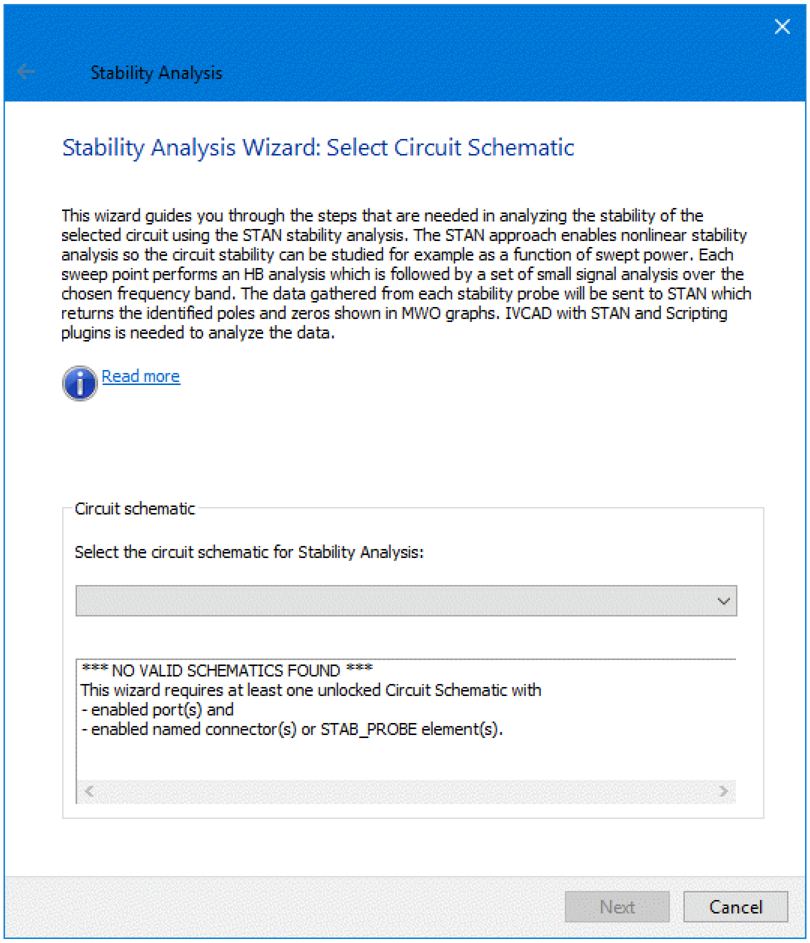
27. You can continue with the VSS program to:
    - consider account impedance mismatch
    - account for frequency dependency
    - run yield analysis
    - perform optimization
    - simulate with modulated signals: EVM, BER, and ACPR
    - replace ideal models with circuits created in the Microwave Office program and/or measured data

13.16. Stability Analysis Wizard

The Stability Analysis Wizard guides you through the steps needed to analyze the stability of a selected circuit using the STAN stability analysis. The STAN approach enables nonlinear stability analysis so you can study the circuit stability, for example, as a function of swept power. Each sweep point performs an HB analysis which is followed by a set of small signal analysis over the chosen frequency band. The data gathered from each stability probe is sent to STAN, which returns the identified poles and zeros shown in Microwave Office graphs. You need at least one schematic with a minimum of one port and one NCONN or STAB_PROBE element to run the Stability Analysis Wizard. NCONN elements are searched only from the top level schematic. If the wizard finds STAB_PROBE elements in some of the sub-schematics, stability measurements are created for them. A copy is made only from the top level schematic. IVCAD must be installed on the localhost or on a remote server.

To access the Stability Analysis Wizard:

1. Open the **Wizards** node in the Project Browser and double-click **Stability Analysis using STAN**. The main Stability Analysis dialog box displays.



2. Select the circuit schematic you want to analyze, then click **Next**. To analyze the data, you need IVCAD with STAN and scripting plug-ins. The Stability Analysis/Simulation Settings dialog displays. For Stability Analysis option details, see "Stability Analysis Wizard Simulation Settings Dialog Box".
3. Click **Finish** to run the STAN analysis. The wizard generates a copy of the original schematic and adds an NLSTABILITY block with specified STAN options and small signal test conditions (or updates parameters if they exist) and replaces the selected port with a swept source port if needed, and then connects new probes to selected NCONNS.
4. The wizard runs a simulation, and the IVCAD server analyzes STAN data. The result is a graph with poles and zeros for each STAB_PROBE element (SchematicName_STAN_RESULTS_ProbeName).

13.16.1. IVCAD Server Installation and Configuration

The Stability Analysis Wizard and the Cadence APLAC® HB simulator search your local IVCAD installation path from the Registry. Only one IVCAD can be installed at a time. The IVCAD installation directory is written to the Registry and it remains unchanged when IVCAD is updated.

- On a 32-bit system: HKEY_LOCAL_MACHINE\SOFTWARE\Maury Microwave\IVCAD
- On a 64-bit system: HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Maury Microwave\IVCAD

The installation directory is the value of the "InstallDir" property.

Your script server must be configured with the following settings before using it from the AWR Design Environment platform:

- Port number (ensure that the port is not blocked by a firewall)
- The output termination character must be set to Linux `\n` after each response, or Windows `\r\n` after each response. The wizard and the server must use the same output termination character.
- Charset ISO-8859-1
- **The option to automatically start the server with the application must be selected.**

You can change local IVCAD server settings only on the IVCAD scripting server. To use the new settings you must restart the server, which saves them to *\Documents\IVCAD\settings\settings.xml* when IVCAD closes. If a local IVCAD installation is not found, the remote server is used. The APLAC simulator reads these settings from the XML file(s) before STAN analysis. If a remote server settings file (*StabilityAnalysisWiz_settings.xml*) is not found, only the local IVCAD settings file is used and defaults are used for the missing settings (timeout).

This wizard saves IVCAD remote server settings to the *StabilityAnalysisWiz_settings.xml* file in the IVCAD settings folder. You should run the wizard at least once to create the file and save the correct settings before STAN analysis can be run on the remote IVCAD server. All IVCAD settings are saved only to the XML file, not to the Microwave Office project file (*.emp*). Changing the settings affects all STAN analyses. The Stability Analysis Wizard writes IVCAD settings to the XML file when you click **Finish**. All other wizard settings such as STAN options, Small signal test conditions, and drive conditions are saved with each project.

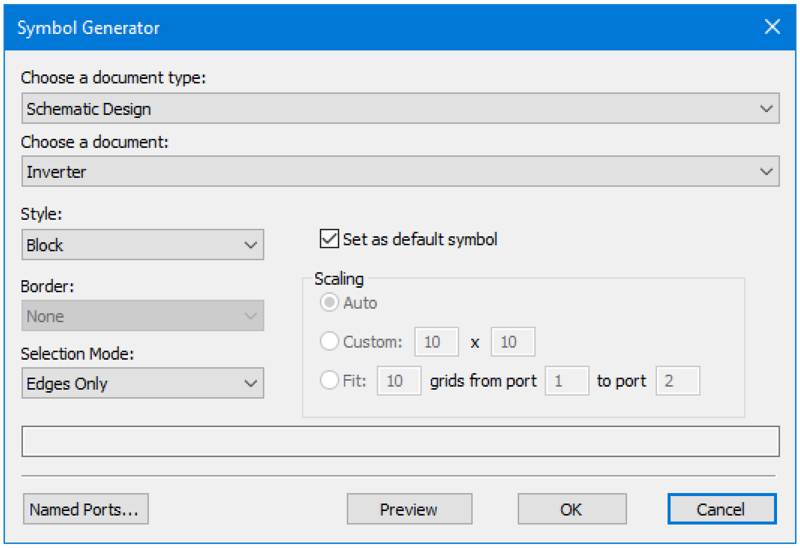
When using an IVCAD remote server, all settings such as port number, output termination character, and charset must be correctly set in the wizard, and you must start the scripting server in IVCAD before performing STAN analysis. If using a remote server, the data is first copied to server's disk and after analysis the result file is copied back to the local disk.

When starting the IVCAD server manually (independent of *awr_as.exe*), you need to set the maximum memory used by IVCAD with a `-m` command line option. For example, `ivcad.exe -m 1000` (where 1000 is max memory in MB, here 1 GB). In the current IVCAD beta, the limit is 1536 MB with a default of 495 MB.

13.17. Symbol Generator Wizard

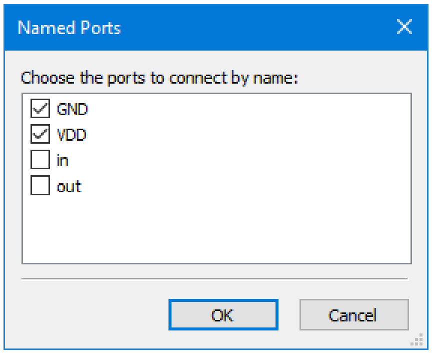
The Symbol Generator Wizard allows you to make custom symbols for subcircuits based on properties of the subcircuit.

To access the Symbol Generator Wizard, open **Wizards** in the Project Browser and double-click **Symbol Generator**.



To use the Symbol Generator Wizard:

1. In **Choose a document type**, select the type of source document you want. (The source defaults to the active window type if it is a supported source.)
2. In **Choose a document**, select the specific source document.
3. In **Style**, select the style for the symbol:
   - **Picture**: The generated symbol looks like the source document.
   - **Block**: The generated symbol is a block with nodes on the appropriate side of the block, depending on the orientation of the ports in the source document.
4. In **Border**, select the border style for the symbol:
   - **None**: No border.
   - **Thin**: Draw a thin border around the bounding box of all the objects in the symbol.
   - **Thick**: Draw a thick border around the bounding box of all the objects in the symbol.
5. In **Scaling**, for **Picture** style symbols select the scaling for the symbol:
   - **Auto**: Creates a symbol based on the size of the current schematic/document display. For schematics and system diagrams this is a 1:1 scaling. For layouts the image is scaled so that the closest two ports are 10 grids apart. For layouts with no ports the image is scaled so that the max side length is 10 grids.
   - **Custom**: Specify (in grid units) the height and length of the symbol.
   - **Fit**: Specify (in grid units) the distance from one specified port to the next.
6. Select the **Set as default symbol** check box if you want to use the new symbol for any new instances of this document type. (Currently only supported for schematics.)
7. If the document utilizes Port_Name elements, click the **Named Ports** button and select the ports you want to display as a named parameter. Unselected ports display as a named pin, as shown in the following figure. If the document does not contain Port_Name elements, the **Named Ports** button is disabled.



Click the **Preview** button at any time during symbol generation to see a preview of the new symbol. Note that the preview window does not close until you close the wizard.

13.18. VSS RF Budget Spreadsheet Wizard

The VSS RF Budget Spreadsheet Wizard (RFB Spreadsheet Wizard) allows you to perform VSS RF Budget Analysis designs in a spreadsheet-like manner. *This wizard runs in VSS software only.* You can add desired RF blocks as either columns or rows of the spreadsheet. You use the rows of the spreadsheet (or columns if you choose to have the blocks in rows) to specify common parameters such as gain, noise figure, and P1dB and desired measurements. When the spreadsheet is analyzed, the measurement results display in the spreadsheet. You can then print the results or export them to a spreadsheet application compatible with Microsoft Excel®.

The RFB Spreadsheet Wizard has several advantages over using a spreadsheet application:

- The Wizard performs all of the computations; you do not have to put together the equations for cascaded gain or noise.
- The Wizard models impedance mismatches, frequency dependent behavior, and mixer image noise.
- The Wizard generates a system diagram that you can modify or use to run RF Inspector and Time Domain simulations (Time Domain simulations require a separate license).

The RFB Spreadsheet Wizard displays in the AWR Design Environment platform if you have the proper license file (VSS-150) to run the wizard. To access the RFB Spreadsheet Wizard, open the **Wizards** node in the Project Browser and double-click **VSS RF Budget Spreadsheet**.

13.18.1. Using the RFB Spreadsheet Wizard

The following sections describe starting, running, and closing the RFB Spreadsheet Wizard.

13.18.1.1. Starting the Wizard

You can run the RFB Spreadsheet Wizard to create a new spreadsheet or to modify an existing spreadsheet.

To create a new spreadsheet, open the **Wizards** node in the Project Browser and double-click the **VSS RF Budget Spreadsheet** .

To modify an existing spreadsheet, right-click the spreadsheet under the **VSS RF Budget Spreadsheet** node in the Project Browser and choose **Edit**.

The RF Budget Analysis Spreadsheet dialog box (main window) displays.

13.18.1.2. Running the Wizard

The main window of the wizard displays VSS RF blocks as columns, and select parameters and measurements as rows. You can add, delete, and move the block columns and parameter and measurement rows. You can also add rows and columns for your own notes.

You can modify parameter values and notes directly in the cells, and edit row and column properties by double-clicking the appropriate row or column header, or by choosing the **Row Properties** or **Column Properties** commands. You can also change the formatting of the individual cells.

When you are ready to perform a simulation, choose **Simulate > Analyze**. The wizard creates a system diagram with the appropriate blocks and parameter settings, creates one or more graphs with the appropriate measurements, and then performs an VSS simulation. After the simulation is complete the results display in the main window.

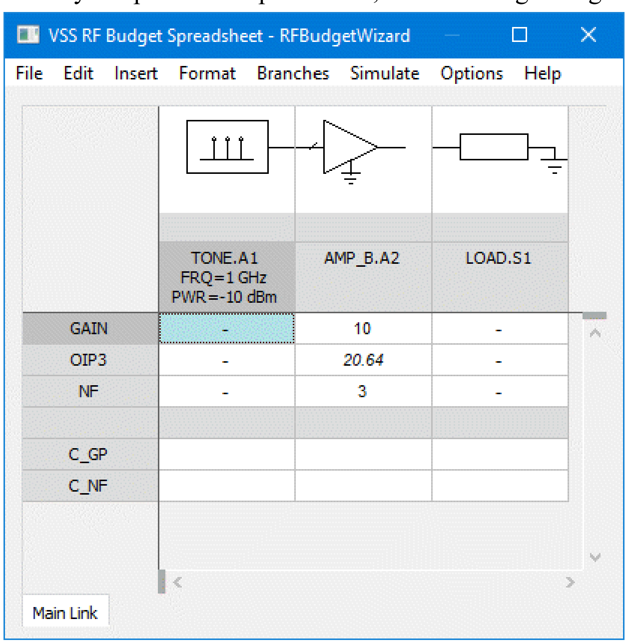
The generated system diagram and graphs remain after you close the wizard, allowing you to use them in other VSS simulations or to modify them directly.

13.18.1.3. Closing the Wizard

To close the RFB Spreadsheet Wizard choose **File > Exit**. If you changed the spreadsheet, the system prompts you to save the changes and may ask if you want to update the generated system diagrams and graphs.

13.18.2. RF Budget Spreadsheet Basics

When you open a new spreadsheet, the following dialog box (main window) displays.



In an RFB spreadsheet, columns typically represent blocks. For block columns, the cells within a column apply to the block indicated at the top of the column.

Rows typically represent common parameters among blocks, such as gain or noise figure, or the desired measurements such as cascaded gain, cascaded noise figure, or power at each output.

You can also add Note rows and columns. A Note row/column allows you to add comments in individual cells, or when blank can serve as spacers.

The top-most row is the header row that normally contains the block symbol, block name, and ID parameter value. It can also contain select parameters and their values, as shown in the TONE.A1 column of the previous figure. In this case, the TONE block's frequency (FRQ) and power (PWR) parameters display.

The left-most column is the header column that normally contains the parameter name for parameter rows, or the measurement name for measurement rows.

13.18.2.1. Display Orientation

By default, the RF blocks display as columns, and parameter and measurements display as rows. You can change this configuration to display blocks as rows, and parameter and measurements as columns by choosing **Options > Options** to display the RF Budget Wizards Options Properties dialog box **General** tab.

This document presumes that RF blocks display as columns. If you choose to display RF blocks as rows, simply exchange the terms "row" and "column" where presented.

13.18.2.2. Cell Selection

Most spreadsheet operations, such as adding or deleting columns or rows, are performed relative to the selected cell or cells. You can select an entire row by clicking the row's header column (the left-most column), or select the entire column by clicking the column's header row.

You can extend the selection to include all cells between the current selection and another cell by pressing the **Shift** key when you click the desired cell.

You can add to or remove from the existing selection by pressing the **Ctrl** key when you click the desired cell. If the cell is not selected, it is selected, otherwise it is de-selected.

13.18.2.3. Block Columns

Individual RF blocks in a design are represented by block columns. Adjacent block columns are considered to have their top-most output port (block to the left) and top-most input port (block to the right) connected. If a block has more than one input or output port, those ports are typically connected to a separate spreadsheet called a Branch.

Adding/Inserting Blocks

To add a new block, first select the desired location by selecting a cell or a column of cells.

- To add the new block before the selected cell, choose **Insert > Insert Block** and then the appropriate sub-command.
- To add the new block after the selected cell, choose **Insert > Add Block** and then the appropriate sub-command.
- You can also right-click on the spreadsheet and choose one of the **Add** commands pertaining to blocks.

After you choose a command, the Block Properties dialog box displays to allow you to select the specific block and configure its parameters.

Editing Blocks

You can edit the block parameters or the block type in the Block Properties dialog box. When applicable, you can also edit parameter values in a parameter row (see “Editing Parameter Values” for details).

To open the Block Properties dialog box:

- Double-click the header row of the desired block's column.
- Select a cell in the desired block's column, then choose **Edit > Column Properties**.
- Select a cell in the desired block's column, then right-click and choose **Column Properties**.

Click the **Parameters** tab on the Block Properties dialog box to edit the block parameter values. You can also select individual parameters to display them in the header row underneath the block's symbol and name/ID by selecting **Display in Header** for the parameter.

If the block has parameters that match the filter criteria for a parameter row, you can edit that parameter directly in the cell of the row.

13.18.2.4. Parameter Rows

Parameter rows display the values of a select parameter from blocks that contain that parameter. For blocks that satisfy the parameter row's criteria, you can edit the parameter value directly within the cell.

Adding, Inserting and Modifying Parameter Rows

To add a new parameter row, first select the desired location by selecting a cell or a row of cells.

- To add the new parameter row above/before the selected cell, choose **Insert > Insert Parameter**.
- To add the new parameter row below/after the selected cell, choose **Insert > Add Parameter**.
- You can also right-click on the spreadsheet and choose **Add Parameter**.

To modify an existing parameter row:

- Double-click the header column of the desired parameter row.
- Select a cell in the desired parameter row, then choose **Edit > Row Properties**.
- Select a cell in the desired parameter row, then right-click on the spreadsheet and choose **Row Properties**.

These commands display the Parameter Definition dialog box, which allows you to specify the parameter row's filtering criteria. There are two types of filter criteria: parameter categories and explicit parameter names.

Specifying a parameter category allows more flexibility than entering a parameter name. If you enter a parameter name, only parameters that exactly match that parameter name are displayed in the parameter row.

Parameter categories, however, match logically related parameters. For example, choosing GAIN displays GAIN (from amplifiers), GCONV (from mixers), and LOSS (from attenuators and other linear blocks) parameters.

Parameter categories can also convert and auto-compute values based on other parameters or a parameter type. For example, for OIP3, if a block does not have a value specified for IP3, but has GAIN and P1DB specified, the OIP3 value is computed from the GAIN and P1DB values. If a value for IP3 is specified, but the IP3TYP parameter is set to IIP3, then the IIP3 value is converted to OIP3 before display. These auto-computed values display in italics to indicate that they are auto-computed.

Editing Parameter Values

If a cell in a parameter row does not contain a '-' notation, you can change the parameter value by either double-clicking the cell or by selecting the cell and then typing. If the cell contains a '-' notation the parameter does not apply to that block, and you cannot edit the cell. Values that display in italics are automatically computed based on the other parameters of the block. You can override these values by typing a value in the cell. For example, in the previous figure, the OIP3 value for the AMP_B.A2 block is automatically computed from the block's GAIN and P1DB parameters (P1DB is not displayed in the spreadsheet, but you can view it by double-clicking the block's header row).

13.18.2.5. Measurement Rows

A measurement row displays the results from a specific RF Budget Analysis measurement in a single row.

To add a new measurement row, first select the desired location by selecting a cell or a row of cells.

- To add the new measurement row above/before the selected cell, choose **Insert > Insert Measurement**.
- To add the new measurement row below/after the selected cell, choose **Insert > Add Measurement**.
- You can also right-click on the spreadsheet and choose **Add Measurement**.

To modify an existing measurement row:

- Double-click the header column of the desired measurement row.
- Select a cell in the desired measurement row, then choose **Edit > Row Properties**.
- Select a cell in the desired measurement row, then right-click on the spreadsheet and choose **Row Properties**.

These commands display the Measurement Properties dialog box. Click the **Measurement Definition** tab to select the desired measurement and to configure its settings. Note that typically only a subset of measurement settings are available. Click the **Graph** tab to determine the name of the graph into which the measurement is placed.

13.18.2.6. Simulation

When you are ready to generate measurement results, choose **Simulate > Analyze**. The RFB Spreadsheet Wizard performs the following steps:

1. Generates a system diagram based on the blocks you configured
2. Generates one or more graphs containing the measurements specified in the measurement rows
3. Runs the simulation command
4. Updates the cells of the measurement rows with values from the measurements of the simulation

By default, the simulation is performed using the impedance mismatch modeling setting from the System Simulator Options dialog box. You can change this behavior on the RF Budget Wizards Options Properties dialog box **General** tab (choose **Options > Options**).

If you generated the spreadsheet by loading an existing system diagram, or if you re-opened a spreadsheet after making changes to its system diagram, the process of generating a new system diagram includes creating a backup copy of the system diagram. The backup copy uses the same name as the original system diagram, with "_Backup" appended to the stem name.

13.18.2.7. Saving

To save your spreadsheet, you must first exit the wizard by choosing **File > Exit**. If prompted to save changes, choose **Yes**. You may also be prompted to update the system diagram. Choose **Yes** to update the system diagram, or **No** to leave it unchanged from the last time **Simulate > Analyze**, **Simulate > Generate System Diagram** or **Simulate > Generate Measurements** was chosen.

After exiting the wizard, you must save the project by choosing **File > Save**.

13.18.2.8. Formatting/Appearances

You can control the formatting and appearance of text in individual cells by selecting the desired cell(s) then choosing **Format > Cells** or by right-clicking and choosing **Format Cells**.

You can also control the formatting and appearance of text in cells belonging to a number of predefined parameter rows and measurements by choosing the appropriate command from the **Format** menu. For example, choosing **Format > Gain Parameter Format** configures the formatting for parameter rows set up as the GAIN parameter category. Choosing **Format > Gain Measurement Format** configures the formatting for measurement rows set to Gain measurements such as C_GP or C_GT.

Format commands display the Format Properties dialog box. This dialog box contains tabs for controlling how numeric values are displayed (see Format Properties dialog box: Numbers tab), for changing the typeface, type styles, type size, text and background colors (see Format Properties dialog box: Font and Selection Font tabs), and for controlling the location of text within the cells (see Format Properties dialog box: Alignment and Selection Alignment tabs).

You can also control the column width and row height. To change a column width, click and drag the right edge of the column in the header row to the desired location. To change a row height, click and drag the lower edge of the row in the header column to the desired location.

13.18.2.9. Notes Columns and Rows

Notes columns and notes rows are columns and rows in which you can enter arbitrary text into the individual cells.

To add a notes column or row, first select the desired location by selecting a cell.

- To add a notes column to the left of the selected cell, choose **Insert > Insert Notes Column**.
- To add a notes column to the right of the selected cell, choose **Insert > Add Notes Column**.
- To add a notes row above the selected cell, choose **Insert > Insert Notes Row**.
- To add a notes row below the selected cell, choose **Insert > Add Notes Row**.

By default, notes rows and columns display their text using the formatting specified when you choose **Format > Notes Format**. You can override the formatting for individual cells by choosing **Format > Cells**.

13.18.2.10. Branches

Branches (mixers, combiners/splitters, quad hybrids, and others) are spreadsheets used to define the blocks connected to an input or an output port of a block with more than one input or output port. Branches are never connected to the first input or output port of the block, as the first input or output port of the block is connected to the adjacent block in the spreadsheet containing the block.

Blocks that support branches connecting to their input or output ports include the mixers, combiners, splitters, quadrature hybrids, RF switches, directional couplers, and circulators.

NOTE: Cadence recommends that you limit the complexity of branches to two or three levels. With more complex branches it is possible that the wizard can fail to properly route the connections between input and output ports when it generates the system diagram.

Adding Branches

There are two ways of adding a new branch:

When you add a mixer block, normally a branch for the LO is automatically created. This branch contains the TONE block for the LO source connected to the LO input port of the mixer.

You can add a mixer block by choosing **Insert > Add Block > Mixer** or **Insert > Insert Block > Mixer**. You can also right-click on the spreadsheet and choose **Add Mixer**. These commands display the Block Properties dialog box. If you select one of the mixer blocks in the Block Properties dialog box and then click **OK**, the Mixer LO dialog box displays. You can enter the frequency and power for the LO source in this dialog box. Clicking **OK** creates a new branch with the name of the new mixer block followed by 'LO', as in "MIXER_B.A5 LO".

For all the other supported multiple-input or multiple-output port blocks, you can create a branch by choosing **Branch > New Branch**. This command displays the Branch Properties dialog box, which allows you to specify the name of the branch, and the properties of the start and end points of the branch. Clicking **OK** creates a new branch with the specified properties.

The Branch Properties dialog box lists the input and output ports of blocks to which its ends may be connected. You can also set the start point to be a TONE source, or the end point to be a LOAD termination.

The branch spreadsheet is created with the same parameter and measurement rows as the main spreadsheet. You can change these rows and add new block columns as desired. Modifications to the rows and columns of the branch spreadsheet do not affect the other spreadsheets.

Navigating Branches

Branches are spreadsheets that are modified just like the main spreadsheet. To navigate to a particular branch, click on the tab with the branch's name at the bottom of the active spreadsheet.

When a branch is the active spreadsheet, you can quickly jump to the block connected to the start of the branch by double-clicking the header row of the first column of the branch, which is labeled **<<START>>**. You can also choose **Branches > Go To Branch > Start Block**.

You can quickly jump to the block connected to the end of the branch by double-clicking the header row of the last column of the branch, which is labeled **<<END>>**. You can also choose **Branches > Go To Branch > End Block**.

When a mixer block is selected, you can choose **Branches > Go To Branch > Mixer LO** to jump to the LO source block in the LO branch of the mixer.

Changing Branches

You can change the start and end points of a branch.

To change the start point connected to an output port of a block, first select the block containing the output port of interest. Next choose **Branches > Connect to Branch** to display the Connect Branch dialog box, which lists the output ports of the selected block and a list of available branches. Note that the first output port of the block is not listed, as that port is always connected to the following block in the spreadsheet containing the block being connected.

If **Include branches in use** is selected, the list of available branches includes branches that are already connected to a block's output port. Selecting a branch in use disconnects the selected branch from the block it was connected to and connects it to the selected port of the active block.

Changing the end point connected to an input port of a multiple input port block is performed in a similar fashion. Select the multiple input port block and choose **Branches > Connect to Branch**.

You can disconnect the start point of a branch from its connected output port, converting the start point of the branch into a TONE block. To do so, first make the branch active, then choose **Branches > Make Source Branch**. This disconnects the start point of the branch from the output port to which it is connected, and replaces it with a TONE source block. Be sure to properly configure the TONE block's frequency and power.

To disconnect the end point from a connected input port, choose **Branches > Make Termination Branch**. This disconnects the end point of the branch from the input port to which it is connected and replaces it with a LOAD termination block.

To rename a branch, choose **Branches > Rename Branch** to display the Rename Branch dialog box.

13.18.2.11. Printing

Choose **File > Print** to display a standard Print dialog box that allows you to select and configure the printer, the number of copies, and whether to print the current spreadsheet or all the spreadsheets.

Choose **File > Page Setup** to configure the page orientation (portrait or landscape), the paper size, and the margins.

Choose **File > Header/Footer** to add printed text at the top and/or bottom of each page in the Header/Footer dialog box. You can enter text to display in the header or footer, or you can add special items such as page numbering, the current date and/or time in a number of formats, and the path, document, or project name.

13.18.2.12. Exporting

You can export spreadsheets as Microsoft Excel 2003 XML Spreadsheets, which can be read by most spreadsheet applications, including OpenOffice Calc.

To export the spreadsheets, choose **File > Export**. The Export to Spreadsheet dialog box displays to allow you to enter the file name and location. All spreadsheets are exported, each spreadsheet displaying as a sheet in the workbook.

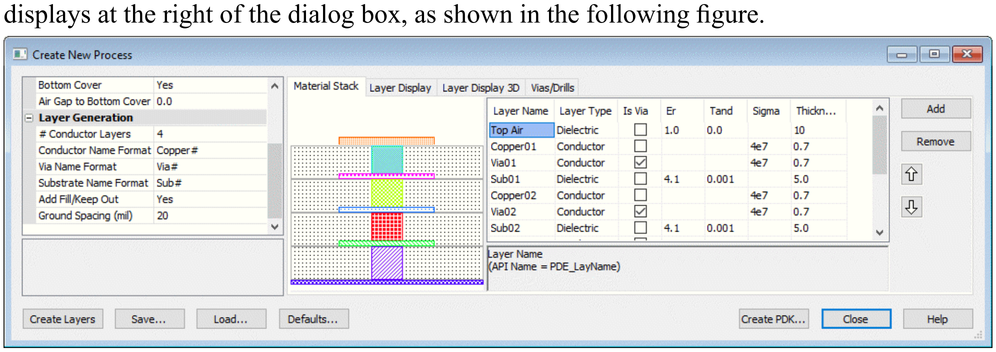
You can also copy spreadsheet cells to the Clipboard and then paste the selected cells into Microsoft Excel or OpenOffice Calc by choosing **Edit > Copy**.

NOTE: When pasting a range of cells into Microsoft Excel, to paste the cell formatting use Excel's **Paste Special** command instead of **Paste**, then choose **XML Spreadsheet** as the format.

13.19. Create New Process Wizard

The Create New Process Wizard provides an easy way to create new PCB/LTCC Process Design Kits (PDKs).

To run this wizard, choose **Tools > Create New Process** to display the Create New Process dialog box. After specifying **Options** and **Layer Generation** settings, click the **Create Layers** button. The data created using the **Layer Generation** settings displays at the right of the dialog box, as shown in the following figure.

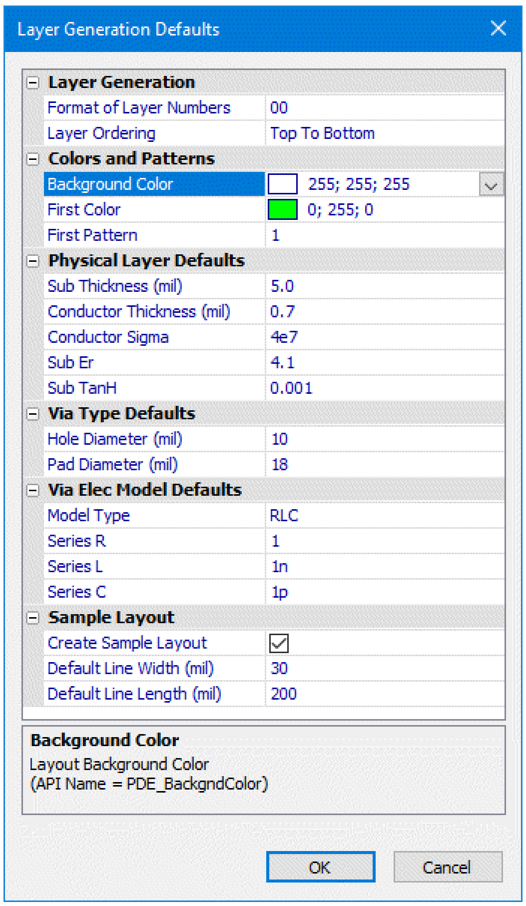


The following tabs display in the dialog box:

- **Material Stack**: Allows you to define and view a representation of new layers, including dielectrics, conductors, and vias.
- **Layer Display**: Allows you to define the layer display. If **Add Fill/Keep Out** is set to "Yes" in the **Layer Generation** settings, +/- layers are created for conductors.
- **Layer Display 3D**: Allows you to specify options for the 3D view of a layout.
- **Vias/Drills**: (Displays with a default example.) Allows you to specify what types of vias are available in the process. Selecting the **Thru** column check box specifies that a single via hole is drilled through all the dielectrics between the selected **Range Start** and **Range End**. If this check box is cleared, then all possible combinations in the specified start/end range are allowed. Clicking in a **Via Model** cell displays a Via Hole Electrical Model dialog box that allows you to edit the Series R, L, and C values, and specify the model type.

See “Create New Process Dialog Box” for details.

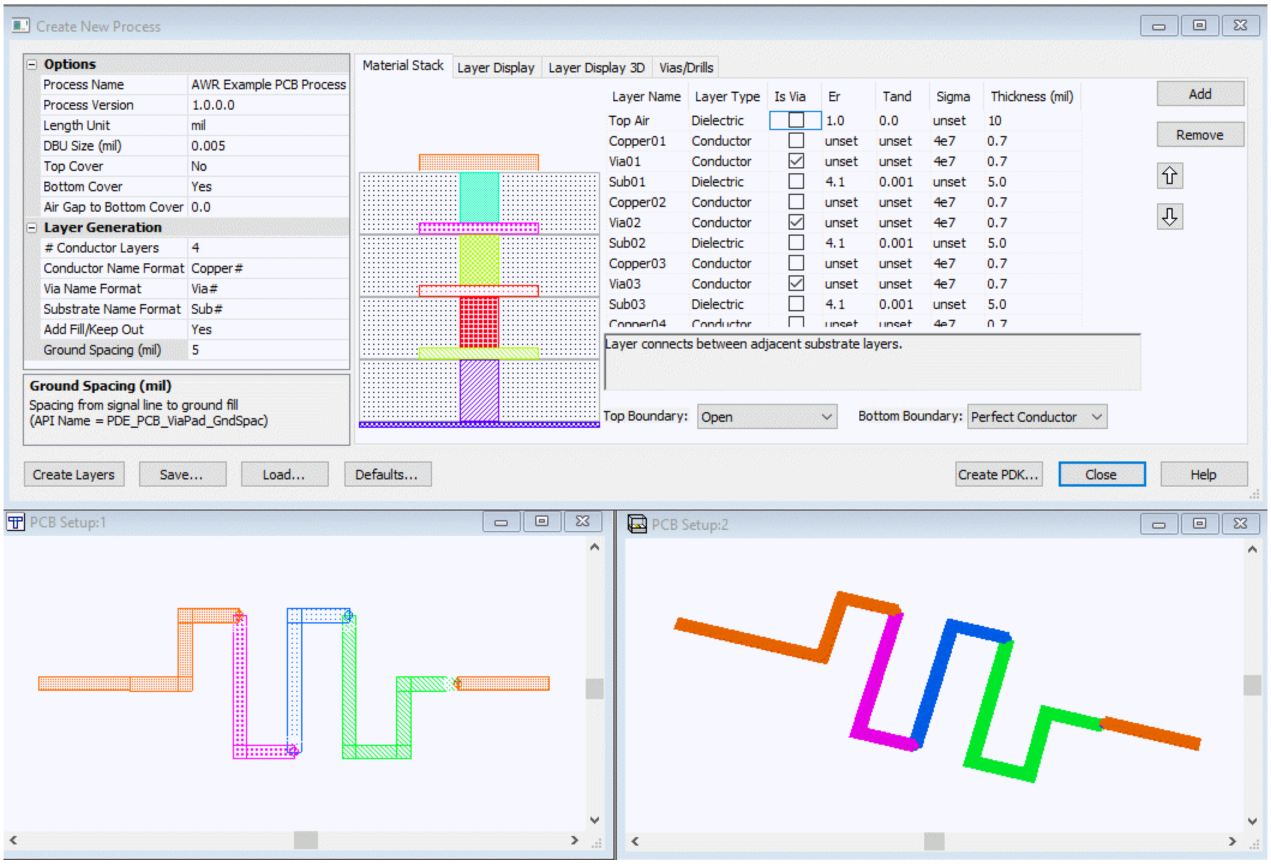
To access additional options for creating layers, click the **Defaults** button to display the Layer Generation Defaults dialog box, as shown in the following figure.



See “Layer Generation Defaults Dialog Box” for details.

After all layer data is correct, click the **Create PDK** button to save the *.ini* file for the new PDK. Standard PDK folders are created in the same folder, along with the LPF file, XML files, template project, and a saved copy (*.xpf* file) of the Process Definition Editor settings that created the kit.

You can load an existing process definition by clicking the **Load** button and selecting a saved *.xpf* file. The current project is reset after prompting you to save any changes, the new PDK and LPF load, via cells are created, and an example layout is generated. The ground flood layers are also added to the example layout; however, by default these layers are off.



A STACKUP is also created and added to the Global Definitions and the XML, with appropriate EM mappings and SPP rules.

13.20. Load Pull Script

The "Load Pull" script allows you to perform load pull simulations on a device model. To run this script choose **Scripts > Load Pull > Load_Pull**. This script has a number of features:

- Contours of the result are within the area of measured impedance points (at the very least within the Smith Chart).
- The ability to edit the impedance points used for load pull on a Smith Chart.
- Additional methods for generating the impedances used in load pull.
- The maximum gamma magnitude increased from 0.95 to 0.9999.
- Multiple algorithms to support load pull point selection and filtering.
- Outputs load pull as a data file in the Microwave Office program in a readable format.
- Simulation points are stored after load pull runs.
- Supports one or multiple simulation sweeps.
- Ability to cancel the simulation while it is running.

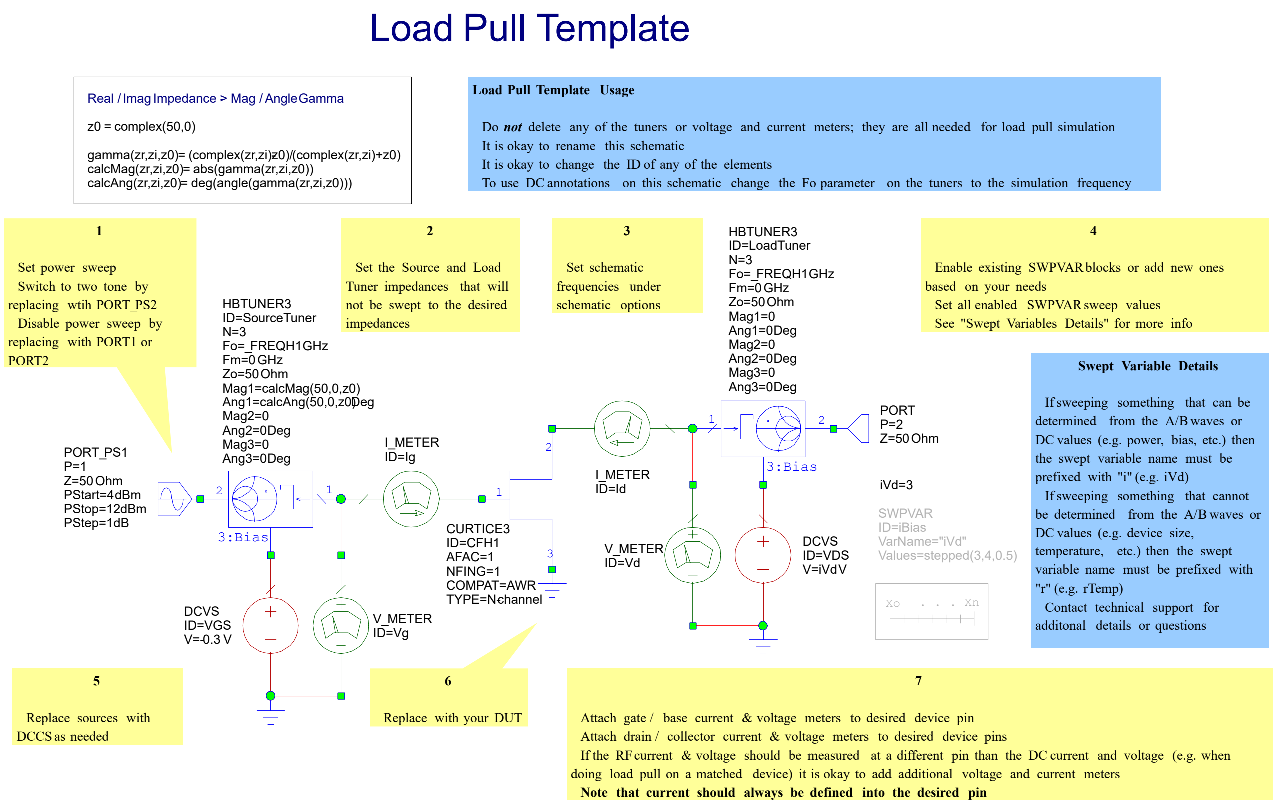
- Observation of the Status Window as the simulation is running.
- Headless mode re-runs load/source pull without any dialog boxes.

13.20.1. Generating a Load Pull Template

Load pull simulation is run from a load pull template schematic that defines the source and load tuners, the voltage and current meters where simulation data is measured, and simulation sweeps.

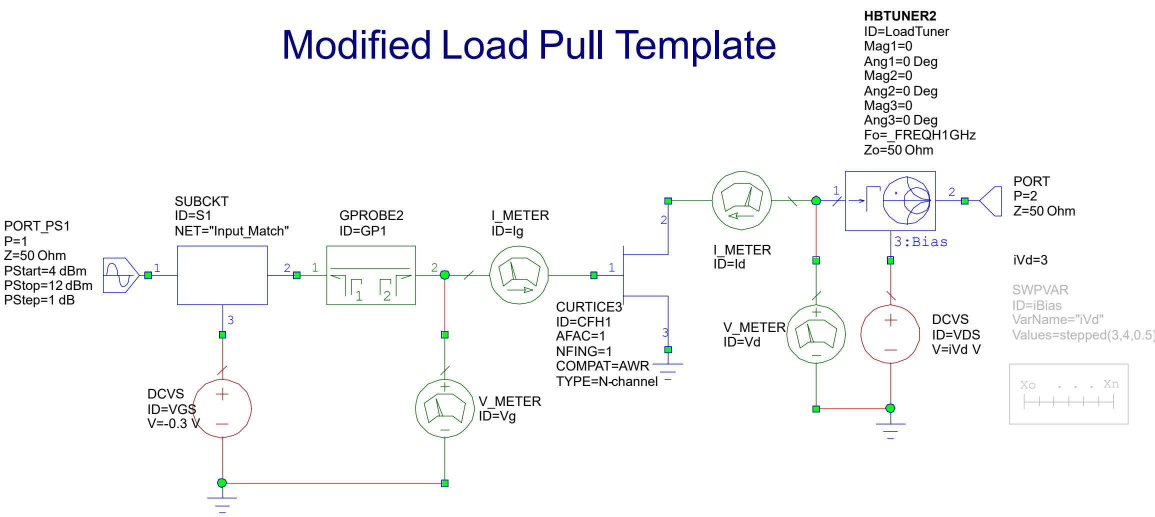
13.20.1.1. Standard Load Pull Template with Two Tuners

To add a template schematic to a project, choose **Scripts > Load Pull > Create Load Pull Template** (if the project already contains a load pull template schematic this script adds an additional template), or choose **Scripts > Load Pull > Load_Pull** to add a template schematic to a project if one does not already exist.



13.20.1.2. Modified Load Pull Template with One Tuner and One Fixed Termination

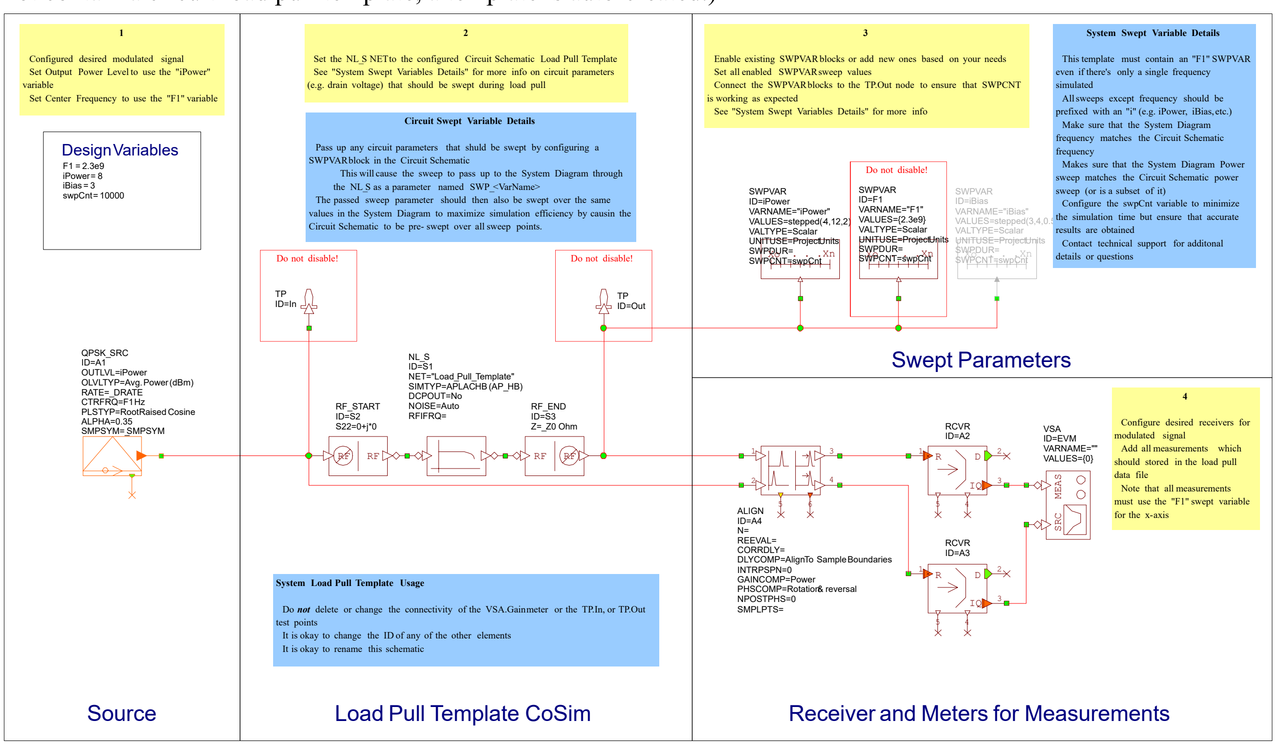
If the source termination needs to be modeled by a fixed and frequency-dependent reflection coefficient for a swept frequency load pull analysis, then the template schematic can be edited to replace the source tuner with a linear network and a GPROBE2 element in series with it. In the following figure, the "Input_Match" SUBCKT could contain, for example, an LPTUNER2 element (with ports 1 and 2 swapped), or a previously designed matching and biasing circuit, or a 2-port s-parameter data file and a BIASTEE in series.



13.20.2. Generating a System Load Pull Template

System load pull simulations are run from a system load pull template that defines the complex source and sinks for any associated measurements along with any system level sweeps.

To add a system template to a project, choose **Scripts > Load Pull > Create System Load Pull Template** (if the project does not contain a circuit load pull template, a template is auto-created.)



13.20.3. Performing Load Pull Simulations

To perform a load pull simulation, choose **Scripts > Load Pull > Load_Pull**. If the project has only one load pull template schematic or if the current active window is a load pull template schematic then the first dialog box to display is the Load Pull Gamma Sweeps dialog box.

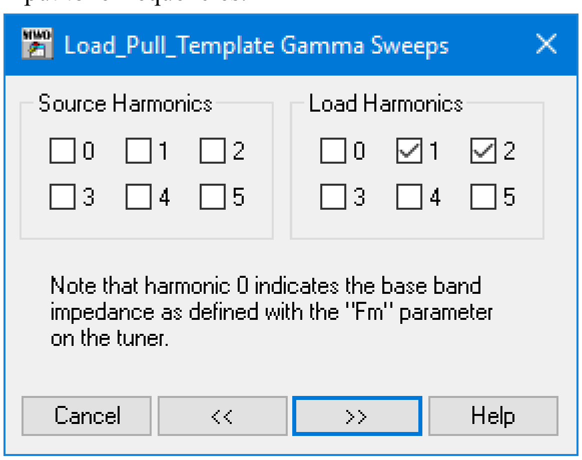
If there are multiple load pull template schematics in the project and the current active window is not one of them, the first dialog box to display is the Choose Load Pull Template Schematic dialog box.

When performing system load pull simulations you should read the template carefully and note the following:

- System level measurements must be set up prior to the simulation.
- Measurements such as ACPR and EVM must plot a single point for each sweep.
- Waveform measurement are not compatible because they need more than one point per sweep.
- The derived values of Output Power, Available Source Power, and Transducer Gain are automatically added to the load pull data file when the load pull simulation is preformed.
- The simulation is run the same as a circuit load pull by choosing **Scripts > Load Pull > Load_Pull**. The rest of the interface is the same as circuit load pull.

13.20.3.1. Load Pull Gamma Sweeps

The Load Pull Gamma Sweeps dialog box is used to select which tuner and harmonics to sweep during the load or source pull. You can select as many check boxes as needed. For example, Load Harmonic 1 is a fundamental load pull only, Source Harmonic 1 is a fundamental source pull only, and Load Harmonic 1 and Source Harmonic 1 are a nested fundamental source and load pull. Harmonic 0 corresponds to intermodulation or baseband frequency for two-tone simulations. To control gamma at low frequencies, set Fm on the desired HBTUNER3 element to the difference between input tone frequencies.



13.20.3.2. Load Pull Gamma Points

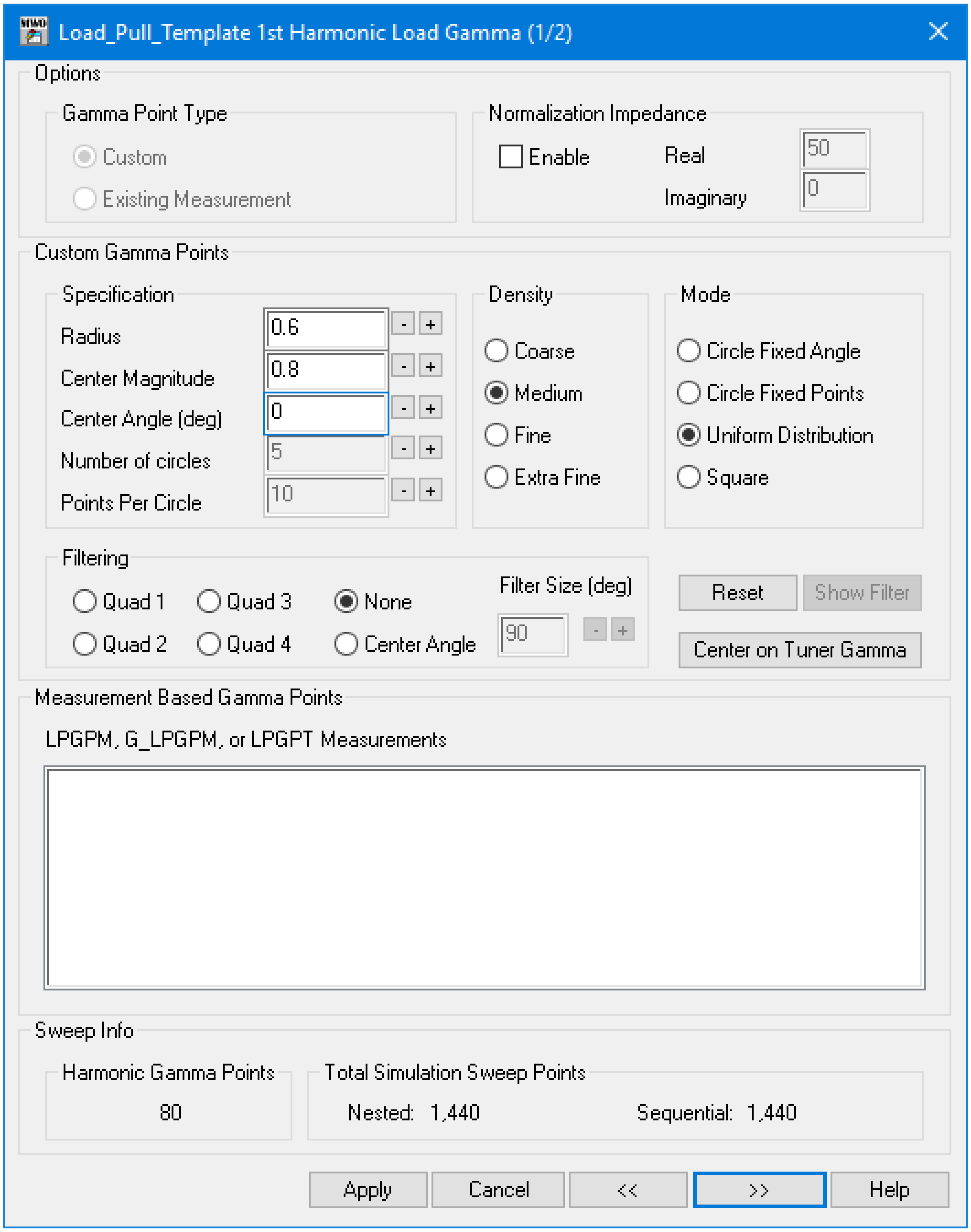
The Load Pull Gamma Points dialog box and associated graph guide you through the load pull gamma point selection. There are two methods for setting up gamma points:

- **Custom** - define your own points with the dialog box controls
- **Existing Measurement** - choose the points from an existing LPGPM, G_LPGPM, or LPGPT measurement (useful for using points defined in an existing load pull data file)

If you choose **Custom** as the **Gamma Point Type** you need to define the load pull points using the controls in the Load Pull Gamma Points dialog box. There are four point computation modes from which to choose:

1. **Circle Fixed Angle** - You control the radius, center of the circles, and number of circles. The number of points on each circle depends on the radius of the circle.
2. **Circle Fixed Points** - You control the radius, center of the circles, number of circles, and number of points per circle. The number of points on each circle is a fixed number.
3. **Uniform Distribution** - You control the radius, center of the circles, and density of the point distribution within the circles.
4. **Square** - You control the radius, center of the squares, and number of rows and columns.

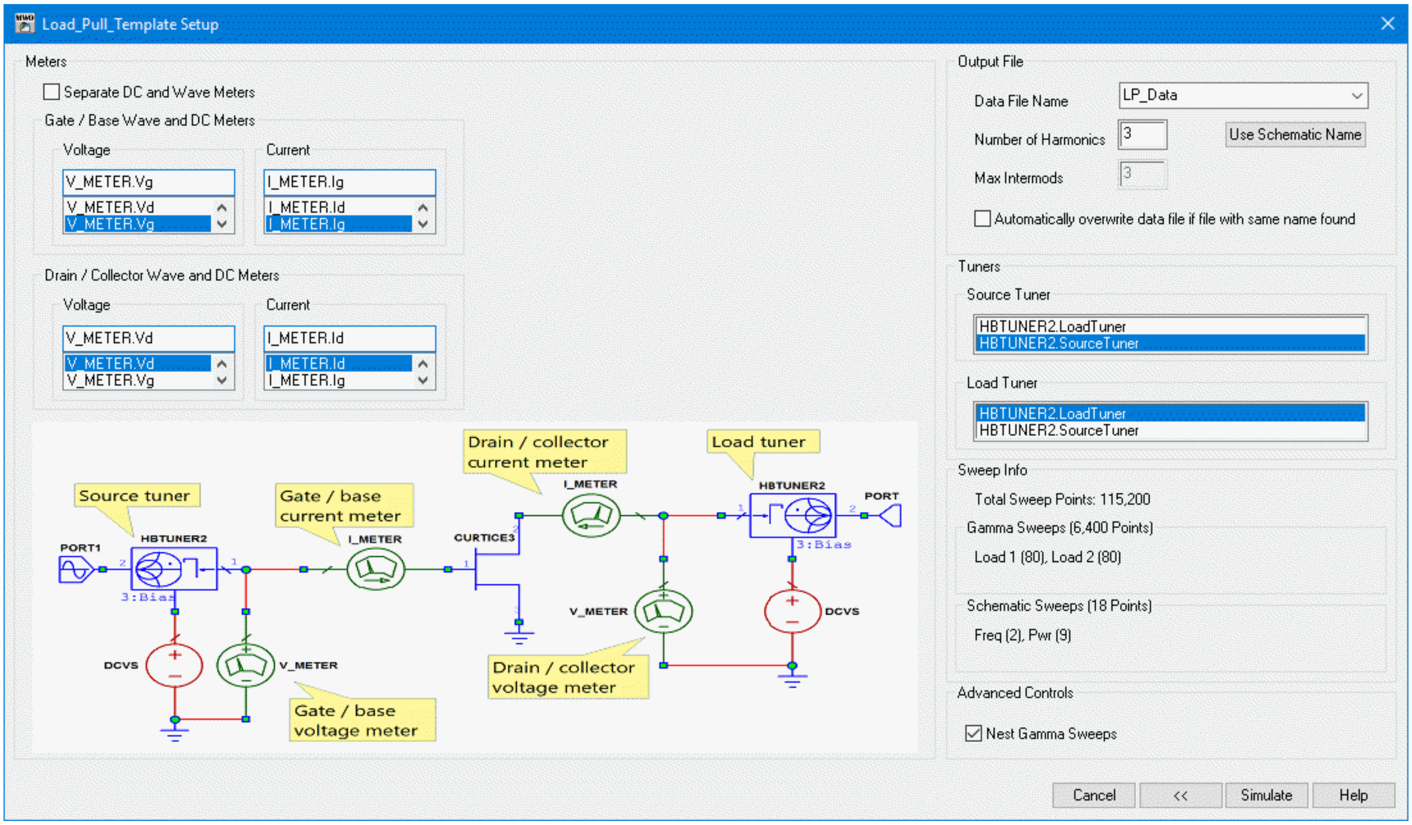
The filtering control allows you to apply a 90-degree window outside of which the points are not used in the load pull analysis. The angle of the filtering window can be set to a fixed quadrant of the Smith Chart or to the angle specified under Custom Gamma Points **Center Angle**. If you choose **Existing Measurement** as the **Gamma Point Type** the load pull data points are taken from the chosen LPGPM, G_LPGPM, or LPGPT. measurement.



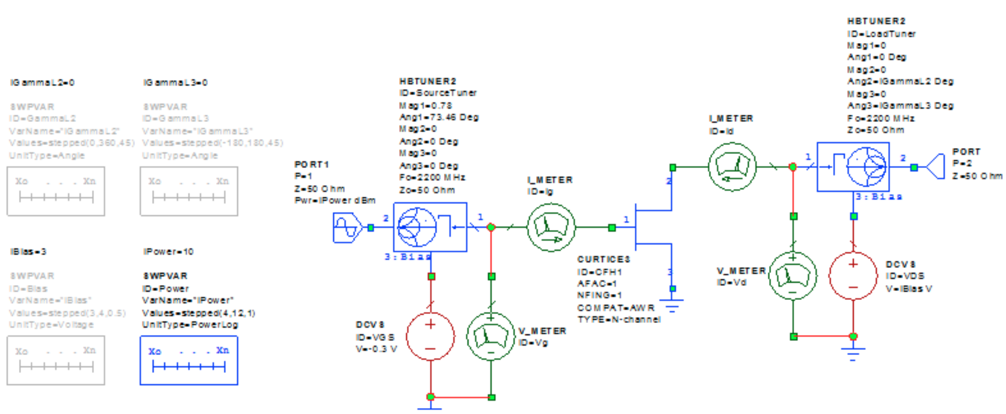
13.20.3.3. Load Pull Setup

In the Load Pull Setup dialog box you select the source and load tuners, the voltage and current meters used to record the load pull data, the name of the data file generated, and the number of harmonics to capture in the data file. If you have not edited the names of the tuners and meters in the load pull template schematic, those fields are correctly configured.

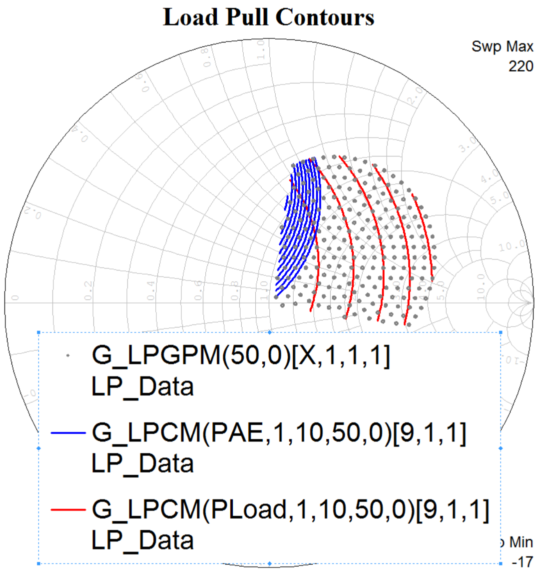
If you modified the template to implement a fixed frequency-dependent source termination, then you must set the **Source Tuner** in this dialog box to the GPROBE2 in series with it. **NOTE: Measurements dependent on available input power (PSrc_Ava) will be invalid if the fixed matching network is lossy.**



After the load pull analysis is complete, you can make measurements on the generated data file. The following is an example of a load pull script result run on the schematic shown. To re-run a load pull without the dialog boxes, right-click the **Load_Pull_Template** node and choose **Options** to display the Options - Load _Pull_Template dialog box. On the **User Attributes** tab, set **Skip Dialogs** to "1". For information on other attributes contact Cadence support.



The following graph shows the simulated gamma points (G_LPGPM measurement) and Output Power and PAE contours (G_LPCM measurement). Obviously these gamma points were not optimally selected for this device, but the mechanics of the load pull simulation and subsequent data plotting are shown.



13.21. Nuhertz Filter Wizard

The Nuhertz Filter Wizard is a third-party filter synthesis program that runs in the Microwave Office program. This wizard displays in the program after you run the Nuhertz filter installer and you have a proper license file (obtained from Nuhertz) to run the wizard. If configured properly the wizard displays in the Project Browser under the **Wizards** node.

To access the Nuhertz Filter Wizard, open the **Wizards** node in the Project Browser and double-click **Nuhertz Filter Wizard**.

To use the Nuhertz Filter Wizard, provide the required information on the **Topology**, **Settings** and **Defaults** tabs. See the Nuhertz documentation for more information about these settings by choosing **Help > Nuhertz Documentation**.

The **Schematic** tab defines what is generated in the Microwave Office software when the filter is synthesized and sent to the AWR Design Environment platform. When you are ready to send to the Microwave Office program, choose **Integration > Send to MWO** .

Chapter 14. Scripts

The Cadence® AWR Design Environment® platform supports an extensive Application Program Interface (API) that allows you to write scripts to solve specific problems. This chapter describes how to use a script after it is developed. For information on developing with scripts in the AWR Design Environment platform see "*Preface*".

Scripts can be Global or (specific to a) Project. A Global script is available in each instance of the AWR Design Environment software you have open. A Project script is only available in the project in which it is loaded. It is easiest and most common to add a new script as a Global script.

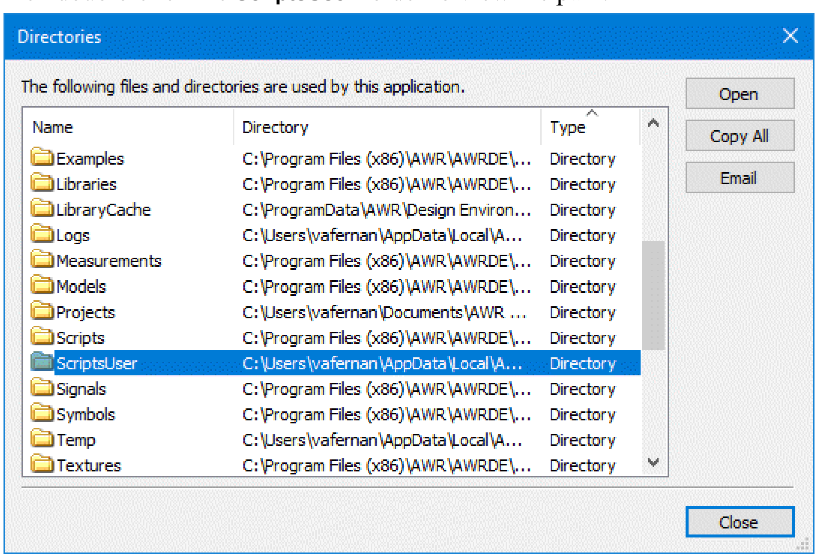
14.1. Running Installed Scripts

The AWR Design Environment platform includes many useful scripts. To access a web page that lists installed scripts with links to their documentation, choose **Scripts > _Show_Help_Pages_For_Global_Scripts > Help**.

The simplest way to run a script is to choose it from the **Scripts** menu. The top of the menu includes subcategories/folders that organize scripts. A category tag is included in each script. See "Running Scripts " in the *API Scripting Guide* for details. The bottom of the menu categorizes as either **Global Scripts** or **Project Scripts** those scripts that do not include category tags.

14.2. Adding a New Script

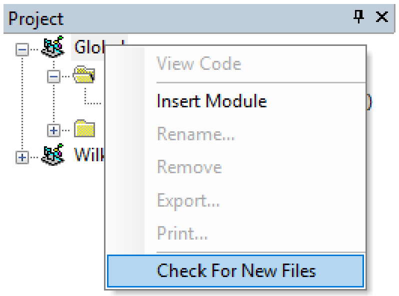
You can choose **Scripts > Configuration > Import_Global_Script** to add a script globally, or you can manually add a script by copying the **.bas* file to your *ScriptsUser* directory. To locate this directory, choose **Help > Show Files/Directories**, then double-click the **ScriptsUser** folder to view the path.



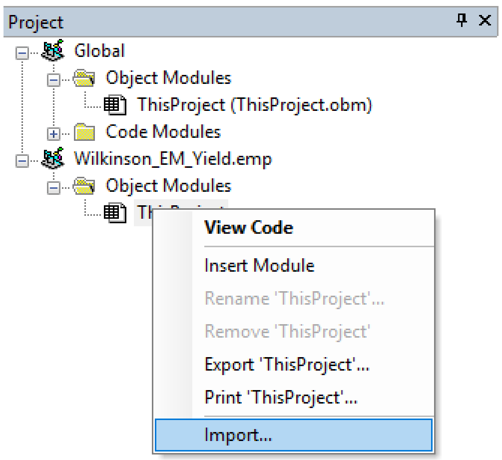
If the AWR Design Environment program is open when you add a new **.bas* file, you must check for new files by opening the Script Development Environment (SDE):

- Choose **Tools > Scripting Editor**, or
- Press **Alt + F11**, or
- Click the **Scripting Editor** toolbar button.

Right-click the **Global** node and choose **Check For New Files**.



You can add a script locally by right-clicking the **ThisProject** node under (project_name).emp, choosing **Import**, and then browsing to the **.bas* file.

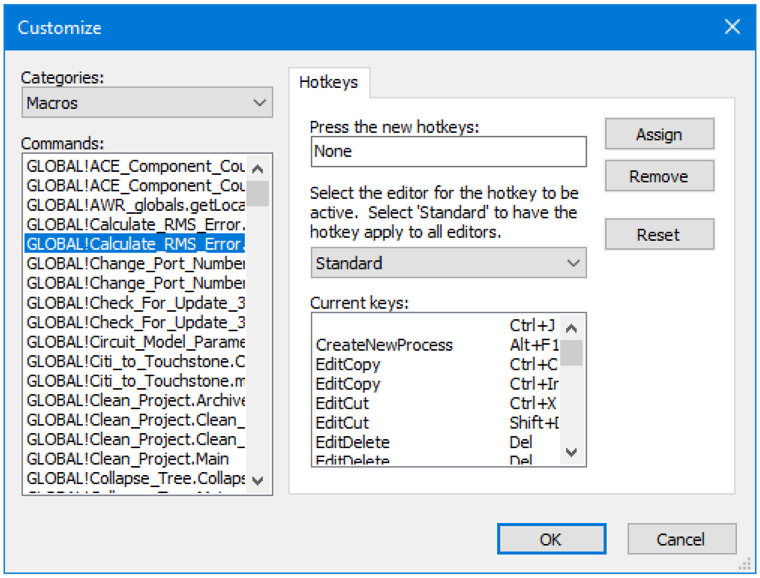


14.3. Customizing How a Script is Run

You can customize the AWR Design Environment platform to call scripts from menus, toolbars, or hotkeys. Customizations are generally applied to Global scripts.

To customize hot keys:

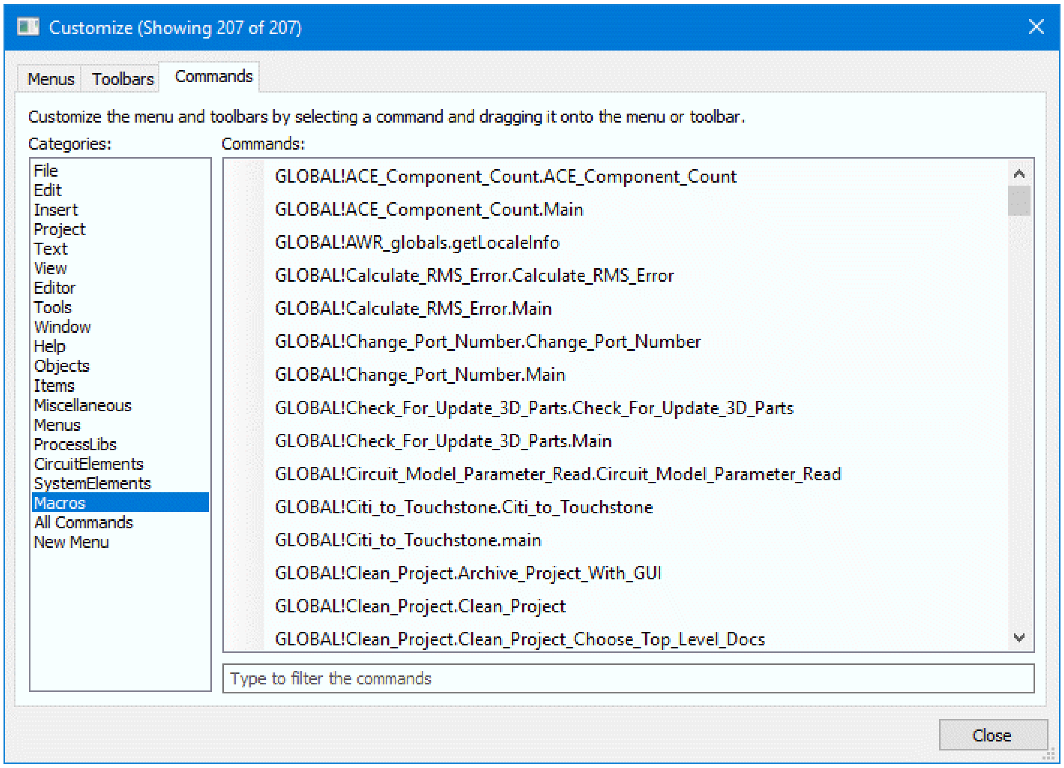
- Choose **Tools > Hotkeys** to display the Customize dialog box.
- In **Categories**, choose **Macros**.



- Select the desired macro.
- Assign the hotkey to the desired editor/view and then click **Assign**.

To customize toolbars or menus:

- Choose **Tools > Customize** to display the Customize dialog box.
- Click the **Commands** tab.
- In **Categories**, choose **Macros**.
- Select the desired macro and drag it across the workspace to the toolbar or menu to which you want to add it, then drop the macro.


Customize (Showing 207 of 207)
Menus
Toolbars
Commands
Customize the menu and toolbars by selecting a command and dragging it onto the menu or toolbar.
Categories:
File
Edit
Insert
Project
Text
View
Editor
Tools
Window
Help
Objects
Items
Miscellaneous
Menus
ProcessLibs
CircuitElements
SystemElements
Macros
All Commands
New Menu
Commands:
GLOBAL!ACE_Component_Count.ACE_Component_Count
GLOBAL!ACE_Component_Count.Main
GLOBAL!AWR_globals.getLocaleInfo
GLOBAL!Calculate_RMS_Error.Calculate_RMS_Error
GLOBAL!Calculate_RMS_Error.Main
GLOBAL!Change_Port_Number.Change_Port_Number
GLOBAL!Change_Port_Number.Main
GLOBAL!Check_For_Update_3D_Parts.Check_For_Update_3D_Parts
GLOBAL!Check_For_Update_3D_Parts.Main
GLOBAL!Circuit_Model_Parameter_Read.Circuit_Model_Parameter_Read
GLOBAL!Citi_to_Touchstone.Citi_to_Touchstone
GLOBAL!Citi_to_Touchstone.main
GLOBAL!Clean_Project.Archive_Project_With_GUI
GLOBAL!Clean_Project.Clean_Project
GLOBAL!Clean_Project.Clean_Project_Choose_Top_Level_Docs
Type to filter the commands
Close

Appendix A. Component Libraries

The Cadence® AWR Design Environment® platform is configured with models, layout cells, and symbols that Cadence develops and maintains. It also includes an extensive library for vendor-specific parts. Often, vendors or customers augment the AWR Design Environment software with their own component libraries by defining the electrical model, the layout cell (Cadence Microwave Office® software only), and the symbol, and then creating XML files that piece all the information together to create a component that can be used in a design.

When defining a component, you must define the following attributes of each component.

For the model, some types are:

- AWR models - Using an existing AWR model and configuring the parameters to model your component.
- File-based models: Using measurement-based model (S-parameter) files or netlists.
- Custom models: Creating your own custom model. This requires additional features and training from Cadence. Contact your local sales manager if interested.

For the layout cells, some types are:

- No layout - Not every element needs a layout, and Cadence Visual System Simulator™ (VSS) communications and radar systems design software does not have layout capability.
- AWR layout cells - Using an existing AWR parameterized layout cell, provided it scales properly for your model.
- Cell library - Using a GDSII or DXF cell library to define a fixed geometry layout.
- Custom layout - Creating your own custom layout cell, typically parameterized. This requires additional features and training from Cadence. Contact your local sales manager if interested.

For the symbols, some types are:

- AWR symbols - Using a symbol from the extensive AWR library.
- Custom symbol - Using the AWR Design Environment platform Symbol Editor to define a new symbol. Note that each new symbol requires additional load time and memory when used, so complicated symbols are not recommended. Additionally, you need to follow the node spacing guidelines for built-in symbols.

A.1. Including Custom Components in the AWR Design Environment

After determining the model, layout cell, and symbol for each part, you generate XML files as the mechanism for the parts that display in the Elements Browser. The specifics of the XML structure and tools to help manage XML files are discussed in later sections. After you have an XML definition, you choose between several mechanisms for including the components-- either using a PDK or including the files in specific folders that the AWR Design Environment software can find.

A.1.1. Using a PDK

A Process Design Kit (PDK) is a configuration of the AWR Design Environment software for a specific foundry process, which is a collection of models, layout cells, symbols and other information. This same mechanism can be used to deliver custom components as well. See “Working With Foundry Libraries” for details on how to use a PDK in a project.

The details for setting up a PDK structure are covered in the training for creating custom models and cells.

The advantages of the PDK approach are:

1. The project stores a reference to the PDK so all users understand what PDK is required to make the project work.
2. The files the PDK uses can be located anywhere on the computer.

The disadvantages of the PDK approach are:

1. A new reference must be added to the project for each PDK, so if there are many, adding them all can take time.
2. The PDK approach is unnecessary when there are no custom models or layout cells.

A.1.2. Using the AppDataUser Folders

You can store your models, cells, symbols, and XML in specified AppDataUser folders to automatically use them, eliminating the need to add a PDK reference to the project.

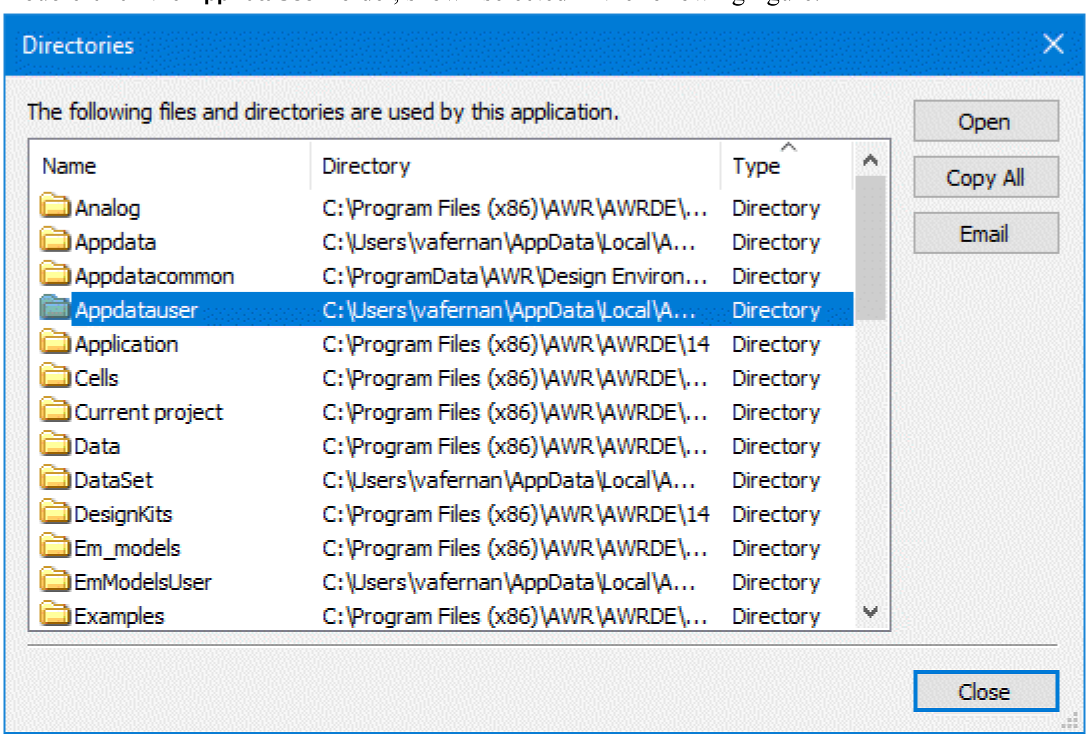
The advantages of the AppDataUser approach are:

1. No references are required with the project. Items in these folders are automatically used.
2. Provides a simple way to add user-defined XML that uses AWR models and layouts.

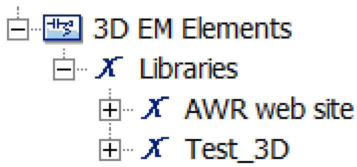
The disadvantages of the AppDataUser approach are:

1. There is no record in the project of where custom models originate, making it difficult if models or cells are missing from a design.
2. The files must be in specific folders on your computer.

With this approach, the data is stored in the AppDataUser folder on the computer. This location can change based on computer settings. To locate this directory, choose **Help > Show Files/Directories** to display the Directories dialog box. Double-click the **AppDataUser** folder, shown selected in the following figure.



In this directory, a folder named *xml* contains three subfolders for the different types of XML: *3D EM Elements* contains 3D parts for Cadence Analyst™ 3D FEM EM simulation, while *Circuit Elements* and *System Blocks* are self-descriptive. Each XML file placed in these folders displays in the AWR Design Environment platform Elements Browser, under the **Libraries** node for that part type, with the same name as the XML file. There can be any number of XML files. There is no need to edit any other XML file to reference these new files. For example, with an XML file named *Test_3d.xml* in the *3D EM Elements* folder, the Elements Browser nodes display as follows:



For models, cells, and symbols, you **must create new folders** under AppDataUser ("next to" the *xml* folder).

- **Models:** for 32-bit AWR Design Environment software, add a folder named *models*, and for 64-bit, add a folder named *models64*. Add your *model.dll* files here for both circuit and system models.
- **Cells:** for 32-bit AWR Design Environment software, add a folder named *cells*, and for 64-bit, add a folder named *cells64*.
- **Symbols:** add a new folder named *symbols* for both 32- and 64-bit AWR Design Environment software.

A.2. Vendor Component Libraries

The **AWR web site** library under the Microwave Office and VSS **Libraries** nodes contains web-based XML component libraries consisting of element or system block models described using an industry-standard XML format posted on the Cadence website. The library contents are automatically retrieved when you click that node in the Elements Browser. If you are not regularly online, you can install a local copy of this library on your computer from the Downloads page of the Cadence website.

NOTES: XML component libraries require Microsoft® Internet Explorer® on your system.

Cadence does not guarantee the accuracy of these vendor libraries. Often Cadence uses S-parameters provided on vendor websites, so it is highly recommended that you check the model to see if it is accurate enough for your needs.

There is no support for using a proxy server to connect to the internet.

A.3. Vendor Library Availability

You can view the list of available Process Design Kits (PDKs) from within the Elements Browser. To see the list, under the **Circuit Elements** node, select **Libraries > *AWR web site > AWR PDK Availability**, then select. a PDK node. Right-click in the bottom pane and ensure **Details** is selected. The current version number is listed as the model name, and the description contains the status of the PDK, the release date, and the version of Microwave Office software supported. The four possible statuses are:

- "Partial" - indicates the PDK is not yet complete.
- "Completed" - indicates the PDK is complete but is not yet tested.
- "Verified" - indicates the PDK is completed and regression tests are in place, but it has not been validated by the foundry.
- "Released" - indicates that the PDK is validated and released by the foundry.

To access Element Help, right-click the PDK in the bottom pane and choose **Element Help** to open (in most cases) the foundry website. With very few exceptions, PDKs are made available from the individual foundries and not through Cadence.

A.4. XML Component Libraries

In the Elements Browser **Libraries** node, Microwave Office software includes support for simple libraries consisting of element models defined by standard Touchstone® (S-parameter) data files and SPICE files. The VSS **Libraries** node in the Elements Browser includes support for libraries consisting of system blocks defined by text files.

Cadence provides a framework for you to generate and maintain your own component libraries. Because Cadence did not do the modeling or take the measured data in the vendor libraries provided, the validity of these libraries is not guaranteed. Vendor libraries are constantly changing, therefore you should decide the best way to model a vendor's parts and build your own libraries.

While the models in simple libraries are adequate for simulation, the models described using XML format include additional information often required to complete a design, such as the model's package, part number, vendor number, and other information.

The XML format is a standard mechanism for sharing database collections over the internet. Database collections described using XML are fully user-extensible and are accessible on a local disk, over an internal network, or shared over the internet.

The following are examples of using XML libraries in the AWR Design Environment platform:

- Setting specific model parameters to AWR models to model a specific vendor part
- Organizing data files with associated part numbers, layout cells, and Help pages
- Creating user-defined folders for frequently used models. XML formatting allows custom folder structuring which you can use to organize the models in the AWR Design Environment platform in any manner.

XML formatting allows you to assign specific values to models built into the AWR Design Environment platform so they accurately model specific library parts. For example, when using a standard resistor model from the Elements Browser **Lumped Elements** category, the resistance value is always set by default. By using the XML format, you can use a resistor with any resistance value directly from the library (you do not have to manually change the resistance setting). Besides being able to set specific AWR model component values, you can use other model formats such as netlists, S-parameter files, and MDIF files.

Using XML libraries also allows you to specify additional information for a library part such as a part number, a layout cell (Microwave Office program only), a symbol, and a Help link (either a file or an HTML link to a Help topic). You can set most of these parameters without XML formatting, but it requires a number of operations. The XML format streamlines all of the settings needed in the AWR Design Environment platform for specific vendor parts, which makes using vendor-specific parts very easy.

This appendix describes the AWR XML schema, discusses working with XML files, and details a technique developed to help generate and manage XML libraries. The technique consists of using a Microsoft Excel® spreadsheet to fill in the library information, and then running a Visual Basic script from within the AWR Design Environment platform to generate an XML file in the proper format.

A.5. AWR's XML Schema Description

The AWR XML schema is a text file that defines the set of keywords (and their attributes) used to describe a component library for use within the AWR Design Environment platform, as well as the rigid hierarchy in which these keywords must be expressed. These keywords, their attributes, and their required hierarchy are presented in the following section.

The AWR XML schema is accessed automatically with the reference *urn:awr-lib-data* at the beginning of the XML.

A.5.1. Keywords, Attributes, and Hierarchy

An XML file that adheres to AWR's XML schema typically contains the following keywords, attributes, and hierarchy (some of which are either mandatory or optional). The following is a sample XML library file:

```
XML_COMPONENT_DATA xmlns=filename
         COPYRIGHT
     SUMMARY
         LPFNAME
     LIBRARYDIR Name=name
     FILE Name=name
     FOLDER Name=name
         FOLDER Name=name
         LPFNAME
         FILE Name=name
            LIBRARYDIR Name=name
                COMPONENT Name=name
                  MODEL
                  ALIAS
                  DESC
                  PARTNUMBER
                  SYMBOL
                  HELP Inline=yes|no
                  CELL
                LPFNAME
                DATA DataType=type Inline=yes|no
                        PARAM Name=name  ReadOnly=yes|no  Hide=yes|no  HideWeak=yes|no
                            LIM
                            STEP
                            TOLP
                            TOLA
                            TOL2P
                            TOL2A
                            DIST
                        OPPOINT
                SUBFILE
                PROPERTY Name=name Value=value OnInstance=yes/no
                NETLISTCMD
                LIBRARYDIR
```

Commonly used keywords and their attributes are described in the following table:

Value	Description
`XML_COMPONENT_DATA xmlns=filename`	Mandatory. The top-level keyword that contains all the data in this file. A file can have only one top-level keyword. `xmlns` is set to the file name containing the schema declaration to which this file adheres. Cadence recommends that

Value	Description
	you use "urn:awr-lib-data" as this always finds the latest *libschema.xml* in the installation directory.
`COPYRIGHT`	Optional. Text string that contains the copyright information for the file.
`SUMMARY`	Optional. Text string that describes the contents of the XML file.
`LPFNAME`	Optional. Text string that specifies the LPF name to associate with the cell library. Useful when working with multi-LPF projects.
`LIBRARYDIR Name=name`	Optional. Defines a library of components. `Name` is set to the name of the library.
`FILE Name=name`	Optional. Points to an XML file that contains a component library. This file is specified as a relative path name or full URL. `Name` is set to the name that displays as a subnode of **Libraries** in the Elements Browser.
`FOLDER Name=name`	Optional. Contains a group of components (`COMPONENT`) or other folders (`FOLDER`). `Name` is set to the name of the folder that displays in the Elements Browser as a subnode of the name specified by `FILE`.
`COMPONENT Name=name`	Mandatory. Defines a component. `Name` is set to the name of the component that displays in the Elements Browser as a subnode of the folder name specified by `FOLDER`. The keywords under the COMPONENT hierarchy must be stated in the following order: `MODEL`, `ALIAS`, `DESC`, `PARTNUMBER`, `SYMBOL`, `HELP`, `CELL`, `LPFNAME`, `DATA`, `OPPOINT`, `SUBFILE`, `PROPERTY`, `NETLISTCMD`, `LIBRARYDIR`.
`MODEL`	Optional. Identifies the simulation model this component uses, such as CAP or RES. You can select any model in the Elements Browser. Note that S-parameters use the SUBCKT model.
`ALIAS`	Optional. Allows creation of a new Elements Browser category.
`DESC`	Optional. Text string that describes this component. This text is displayed in the Elements Browser when the detailed view is selected.
`PARTNUMBER`	Optional. Text string that contains the part number for this component (either defined by the manufacturer or internally-defined).
`SYMBOL`	Optional. Points to a file containing the symbol to use for this component when it displays in the schematic or system diagram window.
`HELP Inline=yes/no`	Optional. Points to a file that contains Help for this component. The file is specified as a relative pathname or full URL. The Help information must be displayable in a web browser; it is typically an HTML or PDF file. If the PDF has internal links, you can point to the link directly by adding a "#" to the end of the PDF name (for example, <HELP>doc1.pdf#Section2</HELP>). `Inline` is set to `yes` or `no` to indicate whether you are specifying data "in-line" or referencing it in a separate file.
`CELL`	Optional. Name of the layout cell, and the *.gds* file name in which it is stored, to be used for this component. The *.gds* file must reside in the same directory as the XML file.
`DATA DataType=type Inline=yes/no`	Mandatory. Contains the data for this model. `DataType` specifies the type of model being defined, and you can set it to one of the values listed following this table. `Inline` is set to `yes` or `no` to indicate whether you are specifying data "in-line" or referencing it in a separate file; however, in-line data is not currently supported.

Value	Description
`PARAM Name=name ReadOnly=yes/no Hide=yes/no HideWeak=yes/no`	Optional. Defines a parameter for this component. There is a `PARAM` entry for each of the component's parameters. `Name` is set to the name of the parameter, such as `C` or `I`. `ReadOnly` is set to `yes` or `no` to indicate whether you can modify this parameter value after the model is placed in a schematic or system diagram. `Hide` is set to `yes` or `no` to indicate whether this parameter displays on the schematic or system diagram and in the Element Options dialog box. `HideWeak` is set to `yes` or `no` to indicate whether this parameter displays on the schematic or system diagram only (it is still available in the Element Options dialog box).
`LIM`	Optional. Contains two values representing the lower and upper limits of the parameter value.
`STEP`	Optional. Contains one value representing the step size of the parameter value.
`TOLP`	Optional. Specifies the tolerance of the parameter as a percentage of the nominal value.
`TOLA`	Optional. Specifies the tolerance of the parameter in absolute terms.
`TOL2P`	Optional. Specifies the second tolerance value for the parameter as a percentage of the nominal value.
`TOL2A`	Optional. Specifies the second tolerance value for the parameter in absolute terms.

`DIST`
 	
`OPPOINT`
	Optional. Type of distribution used for statistical variation for this parameter (`normal`, `uniform`, `discrete`, `log normal`, `normal clipped`, `normal-tol`).
	
	Optional. Specifies Cadence® APLAC® HB simulator operating-point information for AWR Design Environment platform elements defined in the XML. OPPOINT elements are placed within DATA elements.

`SUBFILE DataType=type Inline=yes/no`
 	
`PROPERTY Name=name Value=value OnInstance=yes/no`
	Optional. Specifies a subfile which contains the data for this model. `DataType` specifies the type of model being defined (one of the values listed following this table). For future implementation, `Inline` is set to `yes` or `no` to indicate whether you are specifying data "in-line" or referencing it in a separate file.
	
	Optional. Assigns different properties to a component.`Name` defines the name of the property. `Value` defines the value of the property. `OnInstance` is set to `yes` or `no`. If `yes`, the property displays in the element when you place it in the schematic or system diagram. If `no`, the property is only stored in the XML.

The allowed data types are:

- `sparameter`: Touchstone format S-parameter data.
- `mdif`: MDIF file containing two-port S-parameter data.
- `genmdif`: Generalized MDIF S-parameter data for more than 2-ports.
- `awrnetlist`: File containing AWR netlist.
- `awrmodel`: AWR built-in component model.
- `emstructure`: AWR EM structure model.
- `tsmodel`: File containing Touchstone netlist that is translated to AWR netlist syntax when imported.

- `pspicemodel`: File containing PSpice subcircuit netlist that is translated to AWR netlist syntax when imported.
- `mhmodel`: File containing nonlinear microwave harmonica model.
- `libramodel`: File containing Libra nonlinear model.
- `awrschematic`: File containing an AWR schematic.
- `hspice_trans`: Obsolete
- `hspice_trans2`: File containing an HSPICE netlist that can work with all circuit simulators.
- `aplac_trans`: File containing an APLAC netlist that can work with all circuit simulators.
- `aplac_native`: File containing an arbitrary APLAC netlist that can only use the APLAC simulator.
- `spectre_trans`: File containing a Spectre netlist that can work with all circuit simulators.
- `spectre_native`: File containing an arbitrary Spectre netlist that can only be simulated with Spectre.
- `nmfmodel,paramsdata,nldata` - Reserved for future use.

For a complete list of available keywords, see the AWR schema description in *Library\libschema.xml* in the AWR Design Environment platform installation directory. To obtain use information for data elements not specifically listed in this table contact Cadence AWR Support.

A.6. Creating XML Libraries

To populate the **Libraries** node within the AWR Design Environment platform, you create XML files that adhere to rhw AWR schema described in "AWR's XML Schema Description", and then link them into the application.

While understanding all of XML can be complicated, creating your own XML component libraries is quite simple. When creating XML component library files, you may find the following information helpful:

- The Microsoft Internet Explorer 5.0 XML Validator reads and validates XML files against their schema.
- Other vendors offer tools that allow for data entry and XML generation based on a schema file. The procedures described here are not the only means to generate XML formatted files. With an understanding of the XML format and adequate programming skills, you can generate XML libraries using other means as well.
- The Microsoft "XML Developer Center" available through the Microsoft website contains a wealth of information about creating XML files.
- The AWR schema description is found in *Library\libschema.xml* in the AWR Design Environment platform installation directory.

A.6.1. Creating XML Libraries using XML Files

To create a new XML library:

1. Create a new *.xml* file, or make a copy of one of the files (other than *libschema.xml*) provided in the *\AWR\AWRDE\xx\Library* directory.
2. To define the library, see the allowed keywords and their required hierarchy described in "AWR's XML Schema Description". In addition, see the required syntax shown in "Sample XML File Defining Resistors" and in the *.xml* files in the *\AWR\AWRDE\xx\Library* directory.
3. Begin with the `XML_COMPONENT_DATA` top-level keyword. Under this keyword, create one or more `FOLDER` keywords to contain component models.
4. Under the `FOLDER` keyword, create one or more `COMPONENT` keywords to define the actual component models.

5. Under the `COMPONENT` keyword, create one or more `DATA` keywords to define the actual component data. The `DataType` attribute of the `DATA` keyword specifies the type of component being defined, such as S-parameter data (`sparameter`), AWR built-in component model (`awrmodel`), or Touchstone netlist (`tsmodel`). When defining component data, note the following:

   - When using `DataType=awrmodel`, you must define each parameter in the component model via the `PARAM` keyword. Note that parameter values are specified in MKS; you can use scaling suffixes such as pF to scale values appropriately.
   - When using `DataType=sparameter`, you must provide a reference to a standard Touchstone file that contains the data; this file can be local or specified as a URL.

6. Reference the created XML in the AWR Design Environment platform. See "Using the AppDataUser Folders" for more information.
7. In the AWR Design Environment platform, click the **Elements** tab to display the Elements Browser. The new library is visible as a subnode of the **Libraries** node. Expand the library to view the folders that you defined in your XML library. Expand the folders to see the library's components.

A common requirement is to set a data file's NET parameter to 'read only' as shown in the following example.

```
<DATA DataType="sparameter">
 sparam/r10.s2p
 <PARAM Name="NET" ReadOnly="yes"></PARAM>
</DATA>
```

Another common scenario is to use an S-parameter file in the VSS libraries as shown in the following example.

```
<COMPONENT Name="LP0603A0902AL">
<MODEL>LIN_S</MODEL>
<DESC>0603 Thin Film LPF</DESC>
<PARTNUMBER>LP0603A0902AL</PARTNUMBER>
<SYMBOL></SYMBOL>
<HELP>http://www.avxcorp.com/docs/catalogs/lp0603.pdf</HELP>
<CELL></CELL>
<DATA DataType="awrmodel" Inline="yes">
 <PARAM Name="NET">
  LP0603A0902AL
 </PARAM>
</DATA>
<SUBFILE DataType="sparameter">
  LPF/LP0603A0902AL.S2P
</SUBFILE>
</COMPONENT>
```

A.6.1.1. Sample XML File Defining Resistors

The following is a simple XML file that defines a resistor using various model types.

```
 <?xml version="1.0"?>
<!-- Schema definition; always use this exact line-->
 <XML_COMPONENT_DATA xmlns="urn:awr-lib-data">
 <COPYRIGHT>This library is a copyright of AWR</COPYRIGHT>
 <SUMMARY>This library contains AWR Example Library models </SUMMARY>
```

```
<!-- Define folder for components-->
 <FOLDER Name ="Resistors" Icon ="Resistor">
<!-- Begin component definition-->
  <COMPONENT Name="awr model" >
   <MODEL>REST</MODEL>
   <DESC>10 ohm resist 0.5 percent normal variation</DESC>
   <PARTNUMBER>r1</PARTNUMBER>
   <SYMBOL>Resistor@system.syf</SYMBOL>
   <HELP>HelpEX2/HelpRXAWR1.pdf</HELP>
   <CELL>r10artwork@RcellsGDS.gds</CELL>
   <!-- Example of defining an AWR built-in component model.
Define each parameter via the PARAM keyword-->
   <DATA DataType="awrmodel" Inline="yes">
    <PARAM Name="R" ReadOnly="yes">
     10
     <TOLP>0.5</TOLP>
     <DIST> uniform </DIST>
    </PARAM>
    <PARAM Name="T" ReadOnly="yes">
     25
    </PARAM>
   </DATA>
  </COMPONENT>
<!-- Start component definition-->
  <COMPONENT Name="awr schematic">
   <MODEL>SUBCKT</MODEL>
   <DESC>R1 Schematic</DESC>
   <PARTNUMBER>r1sc</PARTNUMBER>
   <SYMBOL>Resistor@system.syf</SYMBOL>
   <HELP>HelpEX2/HelpRXschem1.pdf</HELP>
   <CELL>r10artwork@RcellsGDS.gds</CELL>
   <!-- Example of defining a model via an AWR schematic.
The schematic is stored relative to this xml file in schem/r10.sch-->
   <DATA DataType="awrschematic">
    schem/r10.sch
   </DATA>
  </COMPONENT>
<!-- Start component definition-->
  <COMPONENT Name="spice netlist">
   <MODEL></MODEL>
   <DESC>r1 spice</DESC>
   <PARTNUMBER>r1sp</PARTNUMBER>
   <SYMBOL>Resistor@system.syf</SYMBOL>
   <HELP>HelpEX2/HelpRXSPICE1.pdf</HELP>
   <CELL>r10artwork@RcellsGDS.gds</CELL>
<!-- Example of defining a model via a PSpice netlist.
The netlist is stored relative to this xml file in spice/r10.cir-->
   <DATA DataType="pspicemodel">
    spice/r10.cir
   </DATA>
  </COMPONENT>
<!-- Start component definition-->
  <COMPONENT Name="sparameter file">
   <MODEL>SUBCKT</MODEL>
   <DESC>R1 Sparameters</DESC>
   <PARTNUMBER>r1sp</PARTNUMBER>
   <SYMBOL>Resistor@system.syf</SYMBOL>
```

```
   <HELP>HelpEX2/HelpRXSparam1.pdf</HELP>
   <CELL>r10artwork@RcellsGDS.gds</CELL>
<!-- Example of defining a model via s-parameter data.
The s-parameter data is stored relative to this xml file in sparam/r10.s2p
This component is setting the s-parameter name to be read only.->
   <DATA DataType="sparameter">
    sparam/r10.s2p
    <PARAM Name="NET" ReadOnly="yes"></PARAM>
   </DATA>
  </COMPONENT>
 </FOLDER>
```

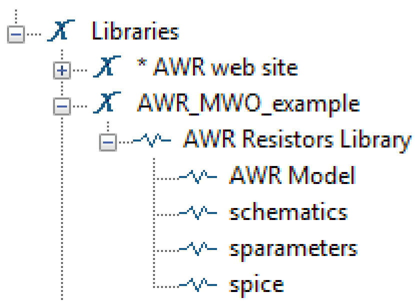
A.6.2. Creating XML Libraries Using Excel Files and Visual Basic

The following sections describe the format needed in a Microsoft Excel file to generate a structurally-correct XML file using the Visual Basic script that Cadence provides. The details of the library are explained and the spreadsheet is demonstrated using an example. The Excel file and script are found at: XML_Library_Generation.zip.

A.6.2.1. Microwave Office Example Library Overview

The Microwave Office example library is a very basic library implemented in four different ways. Figure A.1, "Circuit Element Tree for the AWR Resistors Library" shows the library structure. The library is just three resistors with values of 10, 20, and 30 ohms. The parts from each individual folder display in the lower window of the Elements Browser.

Figure A.1. Circuit Element Tree for the AWR Resistors Library



The Resistors Library is implemented using the AWR REST resistor model, as an AWR schematic file, as an S-parameter file, and as a SPICE netlist. All are equivalent models stored in different formats. The first version uses the AWR REST resistor model and assigns the proper component attributes to it (for example, resistance value, layout, and Help topic). The other three versions are external text files that are loaded into the AWR Design Environment software with appropriate component attributes assigned to them such as layout and Help topic.

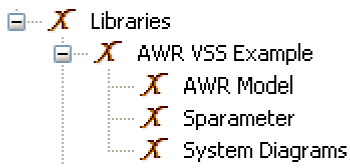
To see the library:

1. Place the *AWR_Resistors_Lib* directory (and its contents) into the *xml/Circuit Elements* folder (choose **Help > Show Files/Directories > XmlUser**).
2. Place *AWR_MWO_example.xml* in the *xml/Circuit Elements* folder.
3. Open a new project to see the **AWR Resistors Library** under the **Libraries** node under **Circuit Elements**.

A.6.2.2. VSS Example Library Overview

This VSS example library is a simple Additive White Gaussian Noise Channel (AWGN) implemented in two different ways, and includes resistors to show how to add S-parameter files as a VSS model. Figure A.2, "System Block Element Tree for the VSS Example" shows the library structure. In the two different AWGN examples, each include three elements with 0, 5, and 10 watts power, and 10, 20, and 30 ohm resistors.

Figure A.2. System Block Element Tree for the VSS Example



The AWGN library was created using the AWR AWGN model and AWR system diagram file. All are equivalent models stored in different formats. The first version uses the AWR AWGN model and assigns the correct component properties to it such as power, powertype and insertion loss. The second one uses an AWR system diagram, where external files are imported into the project.

The resistors library is composed of S-parameter files created using "awrmodel" as the model type and the Cadence AWR LIN_S model as discussed in the following sections.

To view the library:

1. Place the contents of the *AWR_VSS_example* folder into the *xml/System Blocks* folder (choose **Help > Show Files/Directories > XmlUser**).
2. Open a new project to see the **AWR VSS Example** folder under the **Libraries** node under **System Blocks**.

A.6.2.3. Generating the XML Library Using a Visual Basic Script

The previous steps describe how to load example XML libraries into the AWR Design Environment software. Also included with this exercise are the Excel spreadsheets used to generate this library: (*AWR_Resistors.xls* in the *AWR_Resistor_Lib* folder and *AWR_VSS_example.xls* in the *AWR_VSS_example* folder). This file is explained in detail.

Finally, there is an AWR Design Environment platform project file named *Excel_2_XML.emp*. Loaded in this file is a Visual Basic script named *generate_XML_MWO*. When this script runs it asks for an Excel spreadsheet file, processes all of the information in the Excel file, and generates an XML file. The resulting XML file has the same name as the Excel spreadsheet except it has an *.xml* extension. You can run the script on the *AWR_Resistors.xls* file to observe the result.

See "The Script Development Environment" for information on the scripting environment.

A.6.2.4. Excel Spreadsheet Format

Figure A.3, "Blank Excel File for XML Library Generation" shows a generic Excel worksheet with no information set. The column headers help arrange the information.

Figure A.3. Blank Excel File for XML Library Generation

	A	B	C	D	E	F	G
1	Folder	xml model type	parameter name	parameter listing column	parameter value	ICON	
2							
3	COMPONENT	MODEL	DESC	PARTNUMBER	SYMBOL	HELP	CELL
4							
5	End						
6							

There are two major sections to the spreadsheet. Cells A2 through F2 contain specific information for the folder of library parts represented by this worksheet in the Excel file (multiple worksheets are allowed in one Excel file). All of the parts set in an Excel file are made into one XML file, even if there are multiple worksheets. Figure A.4, "Excel File for the Resistors XML Library, Showing Multiple Worksheets" shows a section starting at Row 4 which contains the specific library parts, one part per line.

Figure A.4. Excel File for the Resistors XML Library, Showing Multiple Worksheets

	A	B	C	D	E	F	G	H	I
1	Folder	xml model typ	parameter name	parameter listing column		ICON			
2	AWR model	awrmodel				Resistor			
3	COMPONENT	MODEL	DESC	PARTNUMBER	SYMBOL	HELP	CELL	R	T
4	r1	REST	10 ohm resist 0.5 percent normal variation	r1	Resistor@syste	HelpEX2/HelpRXAWR.doc	r10artw	10 s(0.5p,u)	25
5	r2	REST	20 ohm resistor 0.1 absolute uniform variation	r2	Resistor@syste	HelpEX2/HelpRXAWR.doc	r20artw	20 s(0.2a,n)	37
6	r3	REST	30 ohm resistor 0.1 absolute unform distribution	r3	Resistor@syste	HelpEX2/HelpRXAWR.doc	r30artw	30 s(0.1a,u)	56
7	End								

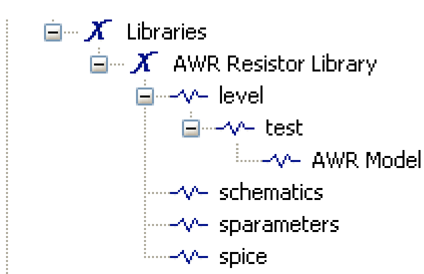
Excel Cell A2 - Folder

The text in cell A2 is the name of the folder used for the data for that specific worksheet. These folder names are subfolders to any folder name given in a higher-level XML file that references this file. For example, in the top level XML file (*AWR_MWO_example.xml*) for this example, there is one line in the file that calls the *AWR_Resistors.xml* file as follows: `AWR_Resistors_Lib\AWR_Resistors.xml`

The first section of this line sets the top level folder name, in this case "AWR Resistors Library". Verify in Figure A.1, "Circuit Element Tree for the AWR Resistors Library" that this is the name of the top level folder for this library. Note that the folder name listed in cell A2 is the name of the subsequent folder name. For example, Figure A.4, "Excel File for the Resistors XML Library, Showing Multiple Worksheets" shows the Excel worksheet for the AWR Model portion of the library. The folder name in cell A2 reads "AWR model" which is the name of the first subfolder for this library.

You can set different levels of folder names by using the "/" character between folder names. For example, if you replace "AWR model" in cell A2 of the spreadsheet shown in Figure A.4, "Excel File for the Resistors XML Library, Showing Multiple Worksheets", with *level/test/AWR model*, the XML tree displays as in Figure A.5, "Example Library with Additional Folder Levels".

Figure A.5. Example Library with Additional Folder Levels



The folder path in cell A2 does not have to be a unique path. You can specify the same folder name for two different Excel worksheets. In this case, all of the parts from both worksheets are placed in the same folder. You can also leave the folder path blank to indicate that all of the parts are located under the top level folder name.

Excel Cell B2 - XML Model Type

The text in cell B2 specifies which XML model type is used for this worksheet. This text must match one of the allowable XML model types. The complete list of available model types is listed in the *libschema.xml* file, which describes the XML format. Some common file types include:

- `sparameter` - an N port S-parameter file
- `awrnetlist` - an AWR netlist file
- `awrmodel` - a model internal to the AWR Design Environment suite
- `pspicemodel` - a SPICE-like netlist for translation into AWR netlist format
- `mdif` - an MDIF file
- `awrschematic` - an AWR schematic file
- `emstructure` - an AWR EM structure file

All of these types reference external files except for awrmodel. When the awrmodel XML model is used, you must specify in column B (the Model column) of the data section which model is used.

For VSS libraries, you only use the awrmodel and awrschematic types.

Excel Cell C2 - Parameter Name

This cell is used when making parameterized xml, and it must be set for each folder that is being parameterized. This text is shown for each parameterized block as the parameter name on the schematic and in the Element Options dialog box. See "Parameterized XML" for more information.

Excel Cell D2 - Parameter Listing Column

This cell is used when making parameterized xml. You specify the column from which you use the text to describe parameterization. See "Parameterized XML" for more information.

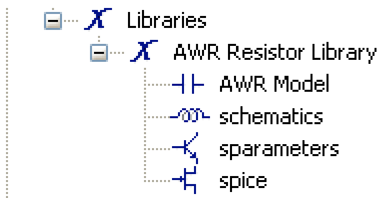
Excel Cell E2 - Top Parameter

This cell is used when making parameterized xml. If you are parameterizing through hierarchies, this cell is used. See "Parameterized XML" for more information.

Excel Cell F2 - Icon

The text in this cell specifies which icon you want for the folder path specified in cell A2. Leaving this blank sets the default icon for the folder. To demonstrate, the example library was changed to use different icons as shown in Figure A.6, "Example Library Using Different Icons". This isn't practical and is for demonstration purposes only.

Figure A.6. Example Library Using Different Icons



The icon name must come from a list of pre-defined icon names included in the following section.

Common XML Icons

The following list includes many of the common Microwave Office icons:

Baluns, Bends, BJT, Capacitor, Coaxial, Coplanar, Coupled Inductor, Coupled Lines, Diode, FET, Filters: Bandpass/Bandstop/Highpass/Lowpass, Inductor, Lines, Lumped Element, Phase, Ports, PwrDivider, Resistor, Resonators, Signal, Substrates, Transformer, Transmission Lines, Waveguide.

The following list includes many of the common VSS icons:

Adders, Amplifiers, Analog-Digital, Bandpass, Bandstop, Behavioral, BER, Binary, BPSK, Channel Encoding, Channels, Coding/Mapping, Constants, Converters, Data Files, Data Type, dB, Decoders, Demodulators, De-multiplexers, Encoders, File Based, Files, Filters, FSK, General Receiver, General Receivers, GMSK, Highpass, Interleaving, Legacy, Logic, Lowpass, Mag/Phase, Math Functions, Math Tools, Meters, Modulated, Signals, Modulation, Modulators, MPSK, MSK, Multiplexers, Multiplexing, Multipliers, Network Analyzers, Noise, OQPSK, PAM, Pi/4 QPSK, Ports, Primitives, Pulse, QAM, QPSK, Random, Receivers, RF Blocks, Serial-Parallel, Signal Processing, Signal Processors, Simulation Based, Source Encoding, Sources, Tracking/Feedback, Transcendental, Transmitters, Waveforms.

Custom icons are currently not supported in the library.

For a full list of the available icons, see "All Available XML Icons".

A.6.2.5. Data Section of the Spreadsheet

The following sections provide information about entering individual models for each particular library, starting on row 4 of the spreadsheet. One model per line is added to the library. As shown in Figure A.7, "Microwave Office Example Library Worksheet Showing Columns A, B, and C" you must specify an "End" tag in column A after the last entry. The script terminates if it cannot find this tag. The following sections describe the contents of each column in the data section of the spreadsheet.

Excel Column A - Component Information

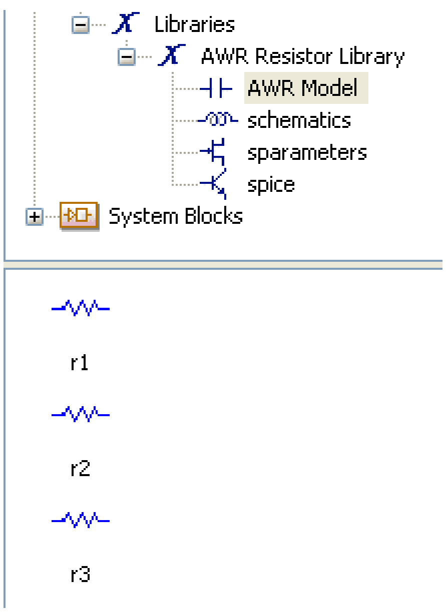
This is the model component section of the library part. The text in this column is the first-level description for the part. When you select a library folder from the XML folder list, the information in column A displays in the lower window of the Elements Browser for each part. For example, Figure A.8, "Example Showing Where Data in Column A Displays in the Library" shows the information in column A for the **AWR Model** folder of the **AWR Resistors Library**. Note the text entered in column A. This figure also shows the elements available in the **AWR Model** folder of the library. Note that the text displayed for each model matches the text in Column A of the spreadsheet. Verify that the other models match with the library in the example project and example spreadsheet provided.

Figure A.7. Microwave Office Example Library Worksheet Showing Columns A, B, and C

	A	B	C
1	Folder	xml model type	parameter name
2	AWR model	awrmodel	
3	COMPONENT	MODEL	DESC
4	r1	REST	10 ohm resist 0.5 percent normal variation
5	r2	REST	20 ohm resistor 0.1 absolute uniform variation
6	r3	REST	30 ohm resistor 0.1 absolute unform distribution
7	End		
8			

Info | AWR model | schematics | spice subcircuit | sparam

Figure A.8. Example Showing Where Data in Column A Displays in the Library

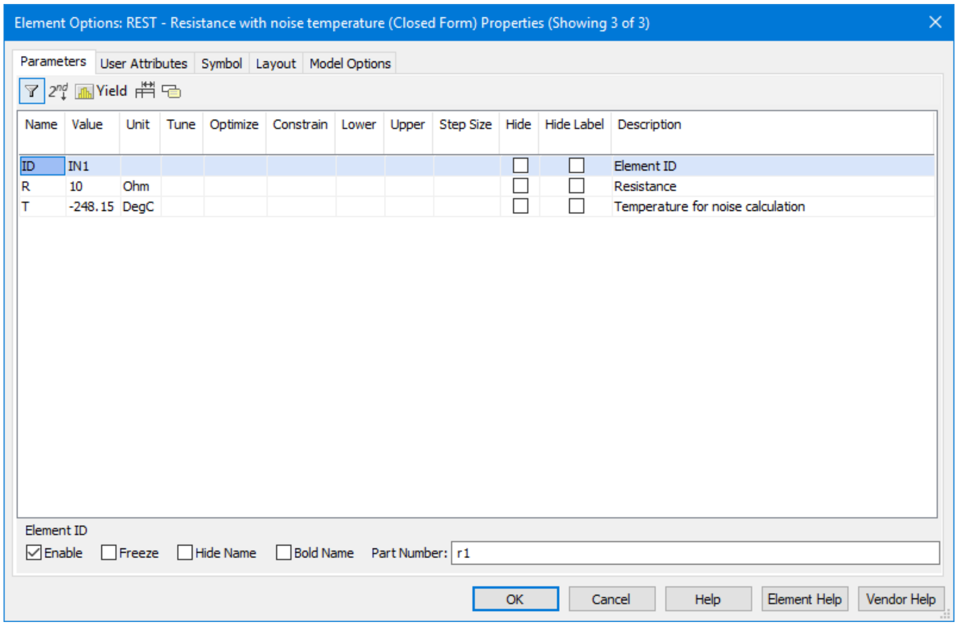


There are different ways to view the parts in a folder (for example, as large icons or small icons). To see the view options, right-click in the lower window of the Elements Browser. Figure A.8, “Example Showing Where Data in Column A Displays in the Library” shows the large icon view.

Excel Column B - Model Name

This column specifies the model in the AWR Design Environment platform to use for the library part. The entries in this column are only necessary if the awrmodel format is specified in cell B2 or if you are making parameterized XML/subcircuit. For all other formats, leave this column blank. These names must match model names in the AWR Design Environment platform. In Figure A.7, “Microwave Office Example Library Worksheet Showing Columns A, B, and C” column B shows that the REST (resistance with temperature) model is used for these library parts. In Figure A.9, “Element Options Dialog Box for REST Model”, one of these library parts is placed in a schematic and the Element Options dialog box is open.

Figure A.9. Element Options Dialog Box for REST Model

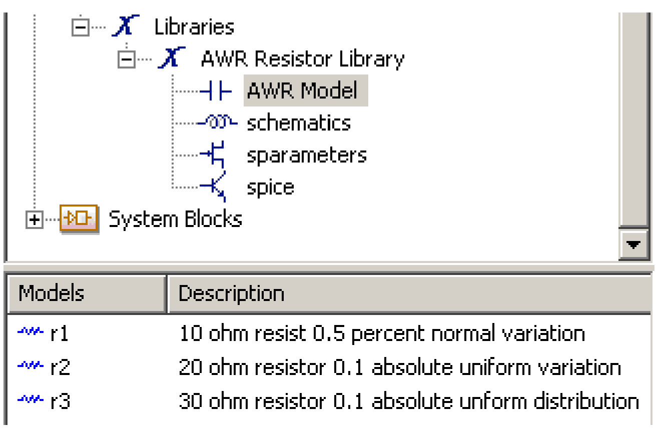


The model name to use in the Excel file is found on the title bar of the dialog box. It is the capitalized name after the "Element Options:" text.

Excel Column C - Model Description

The text in column C is a more detailed description for each part. When you choose a library from the XML folder list, this information displays in the lower window of the Elements Browser when the view is set to **Show Details**. For example, Figure A.7, "Microwave Office Example Library Worksheet Showing Columns A, B, and C" shows the information in column C for the AWR Model version of the resistor library. Figure A.10, "Example of Column C Description Display when Details are Shown" shows the elements available in the AWR Model folder of the library with their descriptions displayed. The description for each model matches the text in Column C of the spreadsheet. Verify the other models match with the library in the AWR Design Environment platform and example spreadsheet provided.

Figure A.10. Example of Column C Description Display when Details are Shown



Excel Column D - Model Part Number

The text in column D is the part number for each model. Each element in a schematic or system diagram (for example a model, S-parameter file, netlist, or EM structure) has a setting for a part number which is useful for a Bill of Materials (BOM) generation for a design or any other user-defined purpose. The Element Options dialog box in Figure A.9, "Element Options Dialog Box for REST Model" shows the **Part Number** at the bottom middle. Figure A.11, "Example Library Worksheet Displaying Columns D, E, and F" shows column D for the part number. Note that the part number specified in Figure A.9, "Element Options Dialog Box for REST Model" matches the text in cell D4 of the example spreadsheet.

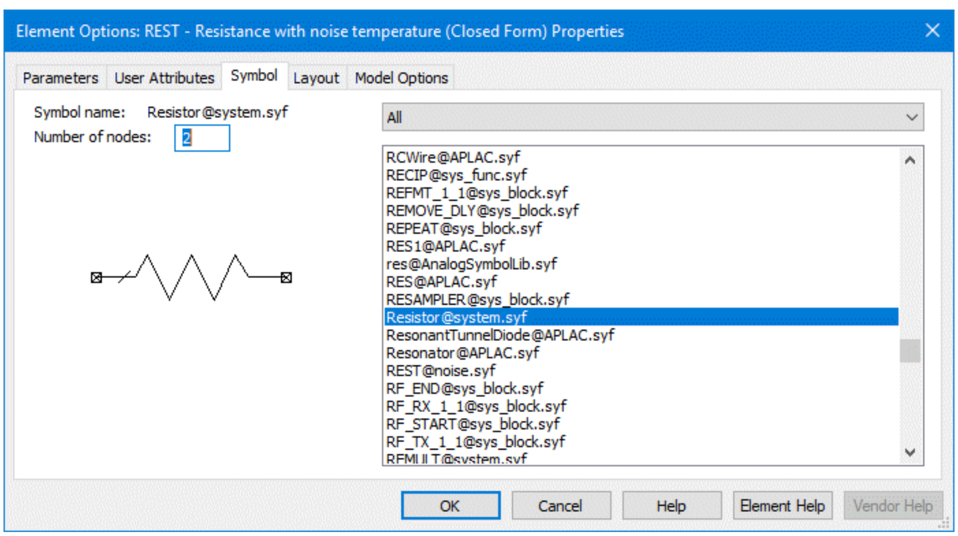
Figure A.11. Example Library Worksheet Displaying Columns D, E, and F

D	E	F
parameter listing column		ICON
		Resistor
PARTNUMBER	SYMBOL	HELP
r1	Resistor@syste	HelpEX2/HelpRXAWR.doc
r2	Resistor@syste	HelpEX2/HelpRXAWR.doc
r3	Resistor@syste	HelpEX2/HelpRXAWR.doc

Excel Column E - Symbol Setting

In the AWR Design Environment platform, you can assign any symbol to any model, including external files such as S-parameter files and netlists. Outside of XML, this assignment is made in the Element Options dialog box on the **Symbol** tab, as shown in Figure A.12, "Element Options Dialog Box Showing Symbol Selection".

Figure A.12. Element Options Dialog Box Showing Symbol Selection



You can use any symbol from the XML library. Every model has a default setting, so this column can be blank. The only requirement is that the number of pins on the symbol must match the number of nodes in the model. It is not possible to assign a 3-pin FET symbol to a resistor model. The best way to determine what to specify in column D is to look at the Element Options dialog box **Symbol** tab, find the symbol you want, and then copy the name of the symbol. This setting is most commonly used to assign a more meaningful symbol to parts that are external files.

The AWR Design Environment platform also allows you to create custom symbols (choose **Project > Circuit Symbols > Add Symbol**). You must manually include a custom symbol file in the *$AWR/symbols* directory. Web-based XML libraries cannot currently download custom symbol files.

Excel Column F - Help Setting

XML libraries can call Help specific to the element library. The Element Options dialog box contains an **Element Help** button and a **Vendor Help** button. Click **Element Help** to open the AWR Help topic for the model from which you opened the dialog box (for example, the REST model in Figure A.12, "Element Options Dialog Box Showing Symbol Selection"). Click **Vendor Help** to open the Help location the vendor specifies, either an HTML URL or an absolute or relative file path. An absolute path contains the entire path to the file, while a relative path to the location of the XML file being read is defined in relation to the current directory.

In the example library, the Help files are Microsoft Word documents located in the *\HelpEX2* folder for the Microwave Office example and the *\Help* folder for the VSS example. In Figure A.11, "Example Library Worksheet Displaying Columns D, E, and F", the settings in column F are relative paths to the Help files for these parts.

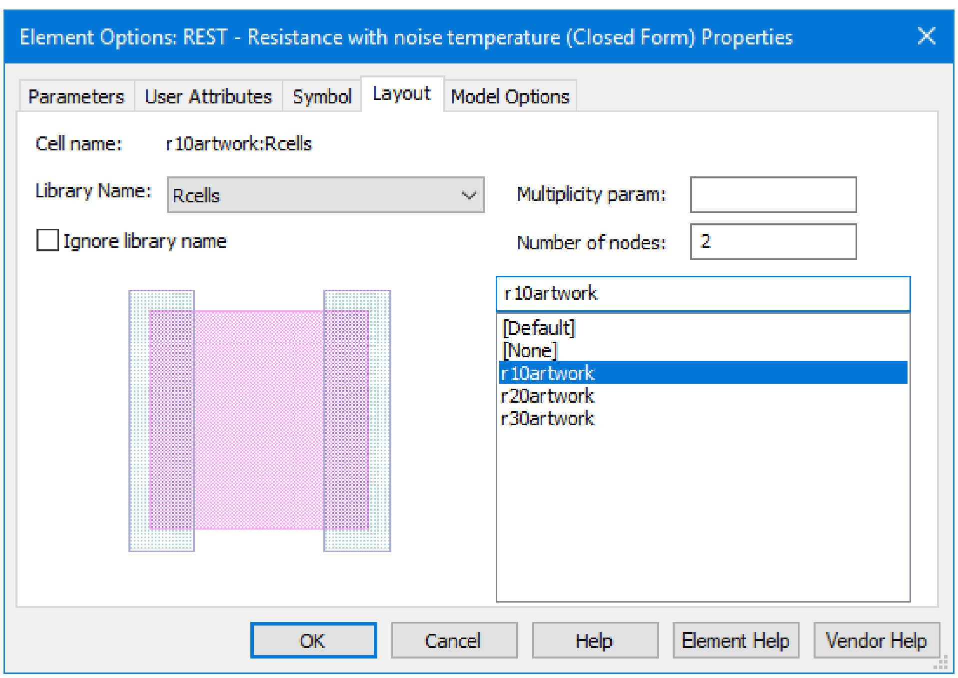
An HTML URL is a more flexible option since website information is easily updated. Web-based Help files can be changed instantly if necessary, and require no changes by library users. Local files require each user to make changes if a Help file changes.

Excel Column G - Layout Cell

Libraries should include layout cells for their parts so Microwave Office library users have the correct layout cell for each part.

Without XML, you select layout cells on the **Layout** tab of the Element Options dialog box, as shown in Figure A.13, "Element Options Dialog Box Showing the Layout Setting". You can assign layout cells that are imported to the Artwork Cell Editor.

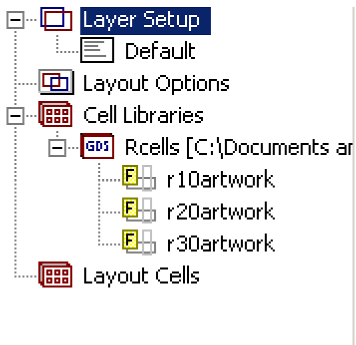
Figure A.13. Element Options Dialog Box Showing the Layout Setting



The syntax for this cell entry is <cellname>@<gdsfilename>. When an XML library part is placed in a schematic, if there is a layout cell specified for that part, the AWR Design Environment software loads the <gdsfilename> GDSII file into the Artwork Cell Editor if it is not already loaded. The GDSII file must be located in the same folder as the XML file that contains this part. The artwork cell specified in <cellname> is assigned as the layout cell for the part.

Figure A.13, "Element Options Dialog Box Showing the Layout Setting" shows the GDSII file used in the example library loaded into the AWR Artwork Cell Editor.

Figure A.14. Layout Manager View with the Example GDSII File Loaded



Notice that under **Cell Libraries** the GDSII library name displays next to the GDS node icon. Next to the library name is the full path to the loaded GDSII file (truncated in this view). Note that the library name **Rcells** is different than the GDSII file name *RcellsGDS.gds*. A list of all cells in the library displays under the library name.

Figure A.15, "Microwave Office Example Library Worksheet Showing Columns G, H, and I" shows how the layout cells are set by listing the cell name and the GDSII file name in the Excel file for this example. Note that the GDSII library name is not used in this syntax.

In the Microwave Office program, a schematic model and a layout cell are associated via cell ports. Cell ports are drawing objects in a layout cell that tell the system how to connect the layout cell to other drawing objects. The cell ports also have numbers which correspond to the node numbers of the schematic element. Although not necessary, it is a good idea to create your GDSII cells so that the number of cell ports matches the number of nodes in the model. If a GDSII file for the library package is not available, you can create it in the Microwave Office program using the Artwork Cell Editor (see AWR Microwave Office Layout Guide for more information). Cadence provides artwork cells in the *$AWR/Examples* folder for some standard packages to help you get started.

Figure A.15. Microwave Office Example Library Worksheet Showing Columns G, H, and I

G	H	I
CELL	R	T
r10artwork@RcellsGDS.gds	10 s(0.5p,u)	25
r20artwork@RcellsGDS.gds	20 s(0.2a,n)	37
r30artwork@RcellsGDS.gds	30 s(0.1a,u)	56

Excel Column H - Column ZZ - Model Information

Model parameter (or file) information is specified starting in column H. Two groups of model types determine the entries in column H; one group is awrmodel and parameterized subcircuit and the other group consists of all other model types.

For all other (non-awrmodel) models, the only model specification is a full or relative path to a data file (for example, a schematic, netlist, or S-parameter) to import into the AWR Design Environment platform, only in column H. Note that S-parameters used in the VSS; program are a special case that is discussed below.

AWR Model Specification

For the awrmodel type, one column is used for each model parameter that is set to a value other than default. This might be only one column for a simple model such as a resistor, or over 100 columns for models such as the BSIM FET model.

For the awrmodel specification, one column is used for each model parameter. The model parameter name is specified in the header column (row 3) and the value is specified in the proper row for that part. The parameter name must match the model parameter name in the AWR Design Environment platform exactly. You can use every column starting with column H, without skipping columns. The script to generate the XML looks at row 3 until it finds a cell that is empty; this is how it determines how many parameters there are.

In the example library for the awrmodel resistors, the temperature resistor model parameters are shown in Figure A.9, "Element Options Dialog Box for REST Model" for one of these parts. Notice that this model has R and T parameters. In the Excel spreadsheet for the awrmodel resistors shown in Figure A.15, "Microwave Office Example Library Worksheet Showing Columns G, H, and I", these model parameters are listed in row 3 of the spreadsheet, and then the assigned values for each model are listed in that parameter's column.

NOTE: If you add an asterisk character (*) to the right of the parameter name (R*), the parameter values under that column cannot be modified in the AWR Design Environment platform. This option is useful when you want to make sure some parameter values of your model remain unchanged.

When entering model parameters, enter the values in base units. For example, enter capacitance in farads, inductance in henries, and current in amps. When a library part is used, these values are converted to the units set in the project.

Notice that there is additional information in the R parameter column. Model parts often have statistical variation to use for yield analysis (also called Monte Carlo analysis). A special syntax for each parameter setting allows statistics to be set for the XML part. To define a parameter for statistical variation, use the following syntax:

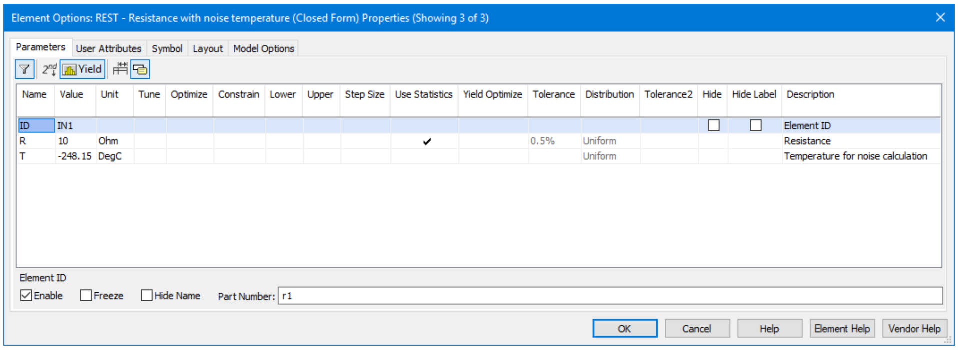
```
<NV> s( <SV><VT>,<ST>)
```

where `<NV>` is the nominal value for the parameter in base units, `<SV>` is the statistical variation either in absolute number in base units or percent, `<VT>` specifies the variation type: `p` for percent or `a` for absolute, and `<ST>` specifies the statistic type: `u` for uniform, `n` for normal, `ln` for log normal, `nc` for normal clipped, or `nt` for normal tolerance.

In Figure A.15, "Microwave Office Example Library Worksheet Showing Columns G, H, and I", the first resistor in the library's **awrmodel** folder has a setting of 10 s(0.5p,u). The nominal value is 10 ohms, and the statistics are 0.5 percent with a uniform distribution. Figure A.9, "Element Options Dialog Box for REST Model" shows the nominal value of R set to 10 ohms. Figure A.16, "Element Options Dialog Box Showing Statistical Settings" shows the statistical settings for this part that display when you click the **Yield** button on the **Parameters** tab toolbar, matching what is in the spreadsheet.

For no statistics, just enter the nominal value as shown for the T parameter in Figure A.15, "Microwave Office Example Library Worksheet Showing Columns G, H, and I"

Figure A.16. Element Options Dialog Box Showing Statistical Settings



AWR Model for VSS LIN_S Model for VSS

There is one special case for VSS libraries. When you want to use an S-parameter file as the model in a VSS simulation, you need to create a library of the awrmodel type using LIN_S as the model, and then specify the path to the S-parameter being used. In this case, the library must load this S-parameter data file also. The script to convert the Excel file to XML can accomplish this. Figure A.17, "Example Library for VSS Using S-parameter Files" shows the Excel data for the VSS library in the example that uses S-parameters as the model.

Figure A.17. Example Library for VSS Using S-parameter Files

	A	B	C	D	E	F	G	H
1	Folder	xml model typ	parameter name	parameter listing column		ICON		
2	Sparameter	awrmodel						
3	COMPONENT	MODEL	DESC	PARTNUMBER	SYMBOL	HELP	CELL	Data
4	r1 SP	LIN_S	20 watts power, 0 T	r1sp		Help\Helpsparam.doc		sparam\r10.s2p
5	r2 SP	LIN_S	35 watts power, 0 T	r2sp		Help\Helpsparam.doc		sparam\r20.s2p
6	r3 SP	LIN_S	50 watts power, 0 T	r3sp		Help\Helpsparam.doc		sparam\r30.s2p
7	End							

AWR Model for VSS LIN_S Model for Microwave Office

Figure A.18, "Microwave Office Example Showing Columns F, G, and H for the S-parameters" shows the Excel library for the S-parameter representation of the AWR Resistors Library for the Microwave Office program. In this library, the S-parameter data is in a subfolder named **sparam**. The column header in cell H3 is DATA. For non-awrmodel parts this header can be named anything.

Figure A.18. Microwave Office Example Showing Columns F, G, and H for the S-parameters

F	G	H
ICON		
Resistor		
HELP	CELL	DATA
HelpEX2/HelpRXSparam.doc	r10artwork@RcellsGDS.gds	sparam/r10.s2p
HelpEX2/HelpRXSparam.doc	r20artwork@RcellsGDS.gds	sparam/r20.s2p
HelpEX2/HelpRXSparam.doc	r30artwork@RcellsGDS.gds	sparam/r30.s2p

The Excel library for the schematic representation of this model is illustrated in Figure A.19, "Microwave Office Example Library Showing Extra Columns for Subfiles". For this library, the schematics are in a subfolder named **schem**. The column header in cell H3 is **DATA**, but you can name this header anything. If the schematics in column H have subfiles like S-parameters, you should include each subfile in the columns next to column H in the same row as that schematic.

You should name the column headers according to the subfiles types: sparameter, awrschematic, awrnetlist, pspicemodel and mdif. For example, if a certain schematic contains three subfiles where two of them are S-parameter files and the other is a schematic, the subfiles for the S-parameters may be included in columns I and J and the subfile for the schematic may be included in column K. There is no preference for which subfile is listed first in the next column after the schematic in column H.

Figure A.19. Microwave Office Example Library Showing Extra Columns for Subfiles

G	H	I	J	K
CELL	DATA	sparameter	sparameter	awrschematic
r10artwork@RcellsGDS.gds	schem/r10.sch	sparam/r10.s2p	sparam/r20.s2p	schem/r30.sch
r20artwork@RcellsGDS.gds	schem/r20.sch			
r30artwork@RcellsGDS.gds	schem/r30.sch			

For the VSS program, the only type used other than the awrmodel type is the awrschematic type. When using this type, you reference system diagrams exported from Cadence that have a **.sys* extension. Figure A.20, "VSS Example Library Showing AWRSCHEMATIC Type for VSS." shows the Excel file for the example VSS library using the awrschematic type.

Figure A.20. VSS Example Library Showing AWRSCHEMATIC Type for VSS.

	A	B	C	D	E	F	G	H
1	Folder	xml model type	parameter name	parameter listing column		ICON		
2	System Diagrams	awrschematic						
3	COMPONENT	MODEL	DESC	PARTNUMBER	SYMBOL	HELP	CELL	DATA
4	AWGN1sys		0 watts power	AWGN1sys	AWGN_1_1@sys_block.syf	Help\Helpsys.doc		sys/AWGN_0.sys
5	AWGN2sys		5 watts power	AWGN2sys	AWGN_1_1@sys_block.syf	Help\Helpsys.doc		sys/AWGN_5.sys
6	AWGN3sys		10 watts power	AWGN3sys	AWGN_1_1@sys_block.syf	Help\Helpsys.doc		sys/AWGN_10.sys
7	End							

A.6.2.6. Optional: Copyright and Summary Settings for 8.0 and older versions

The following lines of an XML file have information about any copyright or summary information. The first three lines of the XML file for this example are shown in Figure A.21, "First Three Lines of the "AWR_Resistors.xml" File"

Figure A.21. First Three Lines of the "AWR_Resistors.xml" File

```
<COPYRIGHT>This library copyright of AWR</COPYRIGHT>
<SUMMARY>This library contains an AWR Example Library models</SUMMARY>
```

You can change these settings by inserting a special worksheet as shown in Figure A.22, "Excel Entries to Set Copyright, Summary and "libschema.xml" Location".

Figure A.22. Excel Entries to Set Copyright, Summary and "libschema.xml" Location

	A	B	C	D	
1	Folder	xml model	parameter	parameter	to
2	../libschema.xml	copyright			
3	This library copyright of AWR				
4	This library containts an AWR Example Library models				

The VB script only uses this information if this worksheet is the first worksheet in the Excel file, so its tab is on the far left. The word "copyright" must appear in cell B2 for this special format to be recognized:

Starting with the version 8.0 AWR Design Environment platform, instead of specifying the location of *libschema.xml* you include the following line in the XML regardless of where the XML file is:

```
<XML_COMPONENT_DATA xmlns="urn:awr-lib-data">
```

The text in cell A3 is copied between the COPYRIGHT tags in the XML file, and the text in cell A4 is copied between the SUMMARY tags in the XML file. This special worksheet is included in the example library.

A.6.2.7. All Available XML Icons

The following is a complete list of all Microwave Office program icons:

AC, Active, Baluns, Bandpass, Bandstop, Bends, BJT, Cap, Capacitor, Coaxial, Components, Coplanar, Coupled Inductor, Coupled Lines, CPW, Data, DC, Dependent, Device Models, Diode, Electrical, EM Models, FET, Filters, Harmonic Balance, Highpass, ICell, Ideal, Inductor, Interconnects, IV, Junctions, Library, Linear, Linear Devices, Lines, Lowpass, Lumped Element, MeasDevice, Meters, Microstrip, Miscellaneous, Negate, Noise, Noise Models, Nonlinear, NonReciprocal, Passive, Phase, Physical, Ports, PRE_RELEASE, Probes, PwrDivider, Resistor, Resonators, RF Elements, Signal, Signals, Signals (Tone 1), Signals (Tone 2), Sources, SPICE, Stripline, Subcircuit, Substrates, Suspended Stripline, Transformer, Transmission Lines, Universal, UPDATED, User Defined, Volterra, Volterra Devices, and Waveguide.

The following is a complete list of all VSS icons:

802.11, 802.11a, Adders, Amplifiers, Analog, Analog Devices, Analog-Digital, Angle, Antennas, Attenuators, Bandpass, Bandstop, Behavioral, BER, Binary, BPSK, CDMA-3G, Channel Encoding, Channels, Coding/Mapping, Combiners/Splitters, Communication Standards, Complex Envelope, Constants, Converters, Couplers, Data Files, Data Type, dB, Decision, Decoders, Demodulators, De-multiplexers, Detectors, Digital Baseband, Dividers, Encoders, File Based, Files, Filters, Fixed Point, Freq. Multipliers, FSK, General Receivers, GMSK, GSM/EDGE, Highpass, Impedance Mismatch, Interleaving, IS-95, Legacy, Linear Filters, Logic, Lowpass, LTE, Mag/Phase, Math Functions, Math Tools, Matrix, Meters, Mixers, Modulated Signals, Modulation, Modulators, MPSK, MSK, Multiplexers, Multiplexing, Multipliers, National Instruments, Network Analyzers, Network Blocks, Noise, OFDM, OQPSK, PAM, Passive, Phase, Phase Noise, Pi/4 QPSK, PLL, Ports, PRE_RELEASE, Primitives, Pulse, QAM, QPSK, Random, Receivers, RF, RF Blocks, Rohde-Schwarz, Serial-Parallel, Signal Processing, Signal Processors, Simulation Based, Simulation Control, Source Encoding, Sources, Switches, Testing, TESTONLY, Tracking/Feedback, Transcendental, Transient, Transmitters, VCOs, Waveforms, and WiMAX Mobile.

A.7. Common XML Library Configurations

There are various ways to build the folder structure for libraries. This section briefly describes several common configurations that you can use.

A.7.1. Configuration 1: Same AWR Model, Different Parameter Sets

This configuration is common for a library of different valued components. Each component uses the same AWR model but has different parameter values set to model each part of the library. This configuration is used in the example library referenced in this document and is identical to Figure A.15, “Microwave Office Example Library Worksheet Showing Columns G, H, and I” where one model is used and the model parameters are listed in rows of the spreadsheet. In this instance, the awrmodel name (column B) must be identical for each component in this folder.

A.7.2. Configuration 2: Multiple AWR Models in One Folder, Using All Default Values

This configuration is used to organize many different AWR models into one folder, and is useful when you use many common parts that you want in one folder, rather than multiple folders throughout the Elements Browser. In this instance, the goal is to have the models in one place, not to set specific values for each model. If no model parameters are set then, use one spreadsheet and specify different model names in column B.

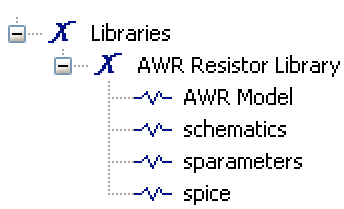
Figure A.23, "Excel Spreadsheet Example for Putting Many Models in One Folder" shows spreadsheet data entry.

Figure A.23. Excel Spreadsheet Example for Putting Many Models in One Folder

	A	B	C	D	E	F	G
1	Folder	xml model typ	parameter name	parameter listing row	top param	ICON	
2		awrmodel					
3	COMPONENT	MODEL	DESC	PARTNUMBER	SYMBOL	HELP	CELL
4	Step	MSTEPX$	Microstrip Line Step Discontinuity				
5	Tee	MTEEX$	Microstrip Tee Discontinuity				
6	Cross	MCROSSX$	Microstrip Cross Discontinuity				
7	90 Bend	MBEN90X$	Microstrip 90 Degree Bend (most accurate)				
8	Bend	MBENDA$	Microstrip Bend				
9	Curve Bend	MCURVE$	Microstrip Curve				
10	Double Radial Stub	MDRSTUB2	Microstrip Double Stub				
11	Series Radial Stub	MRSTUB2	Microstrip Series Radial Stub				
12	Shunt Radial Stub	MSRSTUB2	Microstrip Shunt Radial Stub				
13	Via	VIA	Via				
14	MTRACE	MTRACE2	Microstrip Trace Element, Bends				
15	MCTRACE	MCTRACE	Microstrip Trace Element, Curves				
16	Line	MLIN	Microstrip Transmission Line				
17	Coupled Lines	M2CLIN	Microstrip 2 Coupled Lines				
18	End						

When this library XML file is generated and included in the libraries, the library displays as in Figure A.24, "Elements Browser View of a Library with Many AWR Models". Notice that only columns A, B, and C are used in this mode of operation.

Figure A.24. Elements Browser View of a Library with Many AWR Models

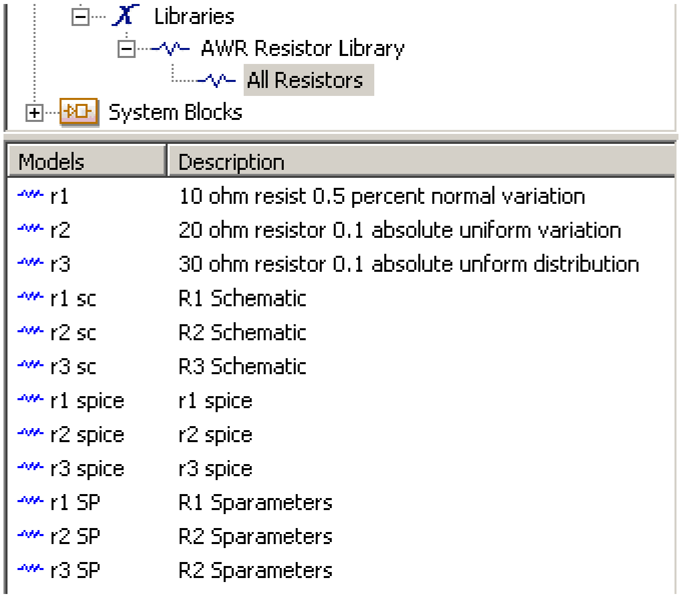


A.7.3. Configuration 3: Multiple XML Model Types in One XML Folder

Sometimes you may want to have many different AWR model types or even different XML model types (for example, awrmodel, sparameter, or awrnetlist) in one folder in an XML library. (This technique also applies when there are just awrmodel types as in configuration 2, but you want to set parameter values and want them all in the same folder.)

Since there can only be one XML model type per spreadsheet, the technique is to break out the different XML model types into different spreadsheets and then assign the same folder to each spreadsheet. For example, in the AWR Resistors library, specify the **All Resistors** folder for all of the models. To do so, cell A2 (the folder cell) must read "All Resistors" on each worksheet in the Excel library file. When this is complete, the library in the AWR Design Environment platform displays as in Figure A.25, "Library with Different Model Types in One Folder".

Figure A.25. Library with Different Model Types in One Folder



A.8. Advanced Options

The following section discusses an advanced option available to component library users.

A.8.1. Referencing Files in XML Files

The AWR XML format allows you to build a library that is a collection of many XML files. Many times when building a large library, you want one grouping of models to be in one XML file and then a top level file that combines all the XML files together to create a complete library of parts. You would typically do this to keep the individual XML files manageable in size. To demonstrate this concept, the Microwave Office example resistor library is altered.

1. Place the *AWR_Resistors_With_Hierarchy* folder in the *xml/Circuit Elements* folder (choose **Help > Show Files/Directories > XmlUser**). When complete, all of the information for this library should be in a *xml/Circuit Elements/AWR_Resistors_With_Hierarchy* folder.
2. In the *$AWR/Library* folder, open the *lib.xml* file and add the following line to the file:

```
<FILE Name="AWR Resistors Library Rev2" Icon="Resistor">
                        AWR_Resistors_With_Hierarchy\AWR_Resistors.xml</FILE>
```

   This line tells the XML libraries to read the *AWR_Resistors.xml* file, which adds this library to the AWR Design Environment platform.
3. Open a new project to see **AWR Resistor Library Rev2** under the **Libraries** node of **Circuit Elements**.

Now you have two identical AWR Resistor libraries implemented in two different ways. In this new library, each library type is composed of separate XML files and they are all pulled together in the *AWR_Resistors.xml* file as shown in Figure A.26, “XML File Calling Other XML Files”.

Figure A.26. XML File Calling Other XML Files

```
<?xml version="1.0"?>
                    <XML_COMPONENT_DATA xmlns="urn:awr-lib-data">

     <COPYRIGHT></COPYRIGHT>
                         <SUMMARY></SUMMARY>
                         <FILE Name="AWR Model"
Icon="Resistor">Resistors\Resistor_AWRmodel.xml</FILE>
                         <FILE Name="Schematics"
Icon="Resistor">Resistors\Resistor_schematics.xml</FILE>
                         <FILE Name="Spice Subcircuits"
Icon="Resistor">Resistors\Resistor_spice_subcircuit.xml</FILE>
                         <FILE Name="Sparameter"
Icon="Resistor">Resistors\Resistor_sparam.xml</FILE>
```

You can view the rest of the files for this library to see how it is assembled.

A.8.2. Adding User Attributes in XML Files

The AWR XML format allows you to assign a user attribute to an element. To do so, you need to include the <PROPERTY> tag in the XML. Make sure you include "User:" in the name for the tag. The following is an example of adding a "Cost:20" user attribute to the MLIN element.

```
<?xml version="1.0"?>
<XML_COMPONENT_DATA xmlns="urn:awr-lib-data">
 <COPYRIGHT></COPYRIGHT>
 <SUMMARY> </SUMMARY>
 <COMPONENT Name="MLIN Test">
  <MODEL>MLIN</MODEL>
  <DESC></DESC>
  <PARTNUMBER></PARTNUMBER>
  <SYMBOL/>
  <HELP></HELP>
  <CELL/>
  <DATA DataType="awrmodel" Inline="yes">
   <PARAM Name="W" ReadOnly="yes"> 40 </PARAM>
   <PARAM Name="L" ReadOnly="yes"> 100 </PARAM>
  </DATA>
  <PROPERTY Name="User:Cost" Value="20"/>
 </COMPONENT>
</XML_COMPONENT_DATA>
```

To make the user attributes read-only, add an asterisk before the name. For example, to make the previous example read only, change the name from **Name="User:Cost"** to **Name="User:*Cost"**.

A.9. Parameterized XML

Parameterized XML is a technique used to create a single model block in the AWR Design Environment platform that chooses from the models defined in an XML library folder. Parameterized XML allows you to use one schematic block to tune or optimize over a library of parts. For example, there is a library with a folder containing a family of chip capacitors that use the CHIPCAP model. The appropriate model parameters are set in the library to properly model the physical capacitors. Without parameterized XML, you need to add each individual capacitor to your design. If you need to vary the capacitance, you have two options-- you can bring in the other parts and see if they are better (time consuming and you are unable to tune or optimize), or you can tune or optimize the capacitance value of the model, and when optimized, find the closest library part (you may be unable to find a part close enough and this approach works only if the model is a built-in model, not an S-parameter file). Using parameterized XML, you can use one block, and specify in the only available parameter which part to use from the folder you choose, as shown in the following figure.

A.9.1. Creating Parameterized XML Using XML Files

There are additional attributes that are set for the COMPONENT and FOLDER elements in the XML to make it parameterized.

1. Param="". This is the identifier for this component; you must set it for each component. This text is used on the parameterized model when choosing which model in the folder to use. If you are building more than one level of folders with parameters, you must also set this for each folder except the top level folder.
2. ParamName="". You must set this for each folder that is being parameterized. This text is shown for each parameterized block as the parameter name on the schematic and in the Element Options dialog box.
3. ParamDefault="". You must set this for each folder this is being parameterized; it is the default component to use when using this model. The text must match one of the Param="" values from one of the components or subfolders of that folder.

The following is an example of parameterized XML. The example in "Using Parameterized XML" utilizes this XML code.

```
<?xml version="1.0"?>
<XML_COMPONENT_DATA xmlns="urn:awr-lib-data">
<COPYRIGHT>This library copyright of AWR</COPYRIGHT>
<SUMMARY>This library contains an AWR Example Library models </SUMMARY>
<FOLDER Name ="sparameters" Icon ="Resistor" ParamName="data_file"
ParamDefault="R1 Sparameters" >
 <COMPONENT Name="r1 SP" Param="R1 Sparameters">
  <MODEL>SUBCKT</MODEL>
  <DESC>R1 Sparameters</DESC>
  <PARTNUMBER>r1sp</PARTNUMBER>
  <SYMBOL>Resistor@system.syf</SYMBOL>
  <HELP>HelpEX2/HelpRXSparam1.pdf</HELP>
  <CELL>r10artwork@RcellsGDS.gds</CELL>
  <DATA DataType="sparameter">
   sparam/r10.s2p
  </DATA>
 </COMPONENT>
 <COMPONENT Name="r2 SP" Param="R2 Sparameters">
  <MODEL>SUBCKT</MODEL>
  <DESC>R2 Sparameters</DESC>
  <PARTNUMBER>r2sp</PARTNUMBER>
  <SYMBOL>Resistor@system.syf</SYMBOL>
  <HELP>HelpEX2/HelpRXSparam2.pdf</HELP>
  <CELL>r20artwork@RcellsGDS.gds</CELL>
  <DATA DataType="sparameter">
   sparam/r20.s2p
  </DATA>
 </COMPONENT>
 <COMPONENT Name="r3 SP" Param="R3 Sparameters">
  <MODEL>SUBCKT</MODEL>
```

```
    <DESC>R2 Sparameters</DESC>
    <PARTNUMBER>r3sp</PARTNUMBER>
    <SYMBOL>Resistor@system.syf</SYMBOL>
    <HELP>HelpEX2/HelpRXSparam3.pdf</HELP>
    <CELL>r30artwork@RcellsGDS.gds</CELL>
    <DATA DataType="sparameter">
     sparam/r30.s2p
    </DATA>
   </COMPONENT>
  </FOLDER>
</XML_COMPONENT_DATA>
```

In parameterized XML, all the parameters inside a component are set to read-only by default, even though all the parameters in the component are set to be parameterized. The `EditableParams` attribute specifies the parameters as not read-only. This attribute needs to be stated at the FOLDER level.

In the following example, the "Scale" parameter is no longer read only.

```
 <FOLDER Name= "RLC_Lowpass" ParamName="RLC_Vals" ParamDefault="R:50,L:1n,C:1p"
EditableParams="Scale">
 <DESC>RLC Lowpass Filter</DESC>
 <SYMBOL>FILTER_LP@sys_block.syf</SYMBOL>
 <COMPONENT Name="RLC" Param="R:50,L:1n,C:1p">
  <MODEL>SUBCKT</MODEL>
  <DESC>RLC Lowpass Filter</DESC>
  <PARTNUMBER>RLC</PARTNUMBER>
  <SYMBOL>FILTER_LP@sys_block.syf</SYMBOL>
  <HELP></HELP>
  <CELL></CELL>
  <DATA DataType="awrschematic" Inline="no" LinkToFile="yes">
   ..\Models\RLC.sch
   <PARAM HideWeak="yes" Name="L">1e-9</PARAM>
   <PARAM HideWeak="yes" Name="R">50</PARAM>
   <PARAM HideWeak="yes" Name="C">1e-12</PARAM>
   <PARAM Name="Scale">1</PARAM>
  </DATA>
 </COMPONENT>
 </FOLDER>
```

A.9.2. Creating Parameterized XML Using Excel Files

This section describes how to enter data in an Excel spreadsheet to create parameterized XML. The Excel spreadsheet for this example is available via the "XML Library Development Application Note" in the AWR Knowledge Base. You can locate this application note and associated files by searching for the term "library development" in the Cadence AWR Knowledge Base. For more information on what the XML looks like, see the preceding sections.

A.9.2.1. Creating Parameterized XML for Microwave Office

Figure A.27, "Excel Spreadsheet Example for Parameterized XML for Microwave Office" This section describes how to fill out Excel spreadsheets to create parameterized XML for the Microwave Office program. The Excel spreadsheet for this example is *AWR_Parameterized_Resistors.xls* in the *AWR_Resistors_Lib* folder. You can create the XML for this Excel spreadsheet by running the *generate_XML_MWO* script in the *Excel_2_XML.emp* project.

The following figure shows how to create the Excel spreadsheet. This example is parameterizing S-parameter files, therefore the Excel spreadsheet is almost identical to the S-parameter library Excel spreadsheet, with the following key differences:

- You must fill out cell A2, the folder name.
- You must fill out cell D2, the parameter column. This is the column whose contents are parameterized.
- The MODEL should be SUBCKT.

Figure A.27. Excel Spreadsheet Example for Parameterized XML for Microwave Office

	A	B	C	D	E	F	G	H
1	Folder	xml model	parameter name	parameter	top param			
2	parameterized	sparameter	Resistance	A				
3	COMPONENT	MODEL	DESC	PARTNUMBER	SYMBOL	HELP	CELL	DATA
4	r1 SP	SUBCKT	R1 Sparameters	r1sp	Resistor@system.syf	HelpEX2/HelpRXSparam.doc	r10artwork@RcellsGDS.gds	sparam/r10.s2p
5	r2 SP	SUBCKT	R2 Sparameters	r2sp	Resistor@system.syf	HelpEX2/HelpRXSparam.doc	r20artwork@RcellsGDS.gds	sparam/r20.s2p
6	r3 SP	SUBCKT	R2 Sparameters	r3sp	Resistor@system.syf	HelpEX2/HelpRXSparam.doc	r30artwork@RcellsGDS.gds	sparam/r30.s2p
7	End							

A.9.2.2. Creating Parameterized XML for VSS

Figure A.28, "Excel Spreadsheet Example for Parameterized XML for VSS" This section describes how to fill out the Excel spreadsheet to create parameterized XML for the VSS program. The Excel spreadsheet for this example is *AWR_VSS_example.xls* on the *Parameterized* worksheet in the *AWR_Resistors_Lib* folder. You can create the XML for this Excel spreadsheet by running the *generate_XML_MWO* script in the *Excel_2_XML.emp* project.

The following figure shows how to create the Excel spreadsheet. This example is parameterizing S-parameter files, therefore the Excel spreadsheet is almost identical to the S-parameter library Excel spreadsheet, with the following key differences:

- You must fill out cell A2, the folder name.
- You must fill out cell D2, the parameter column. This is the column whose contents are parameterized.
- The MODEL should remain LIN_S.

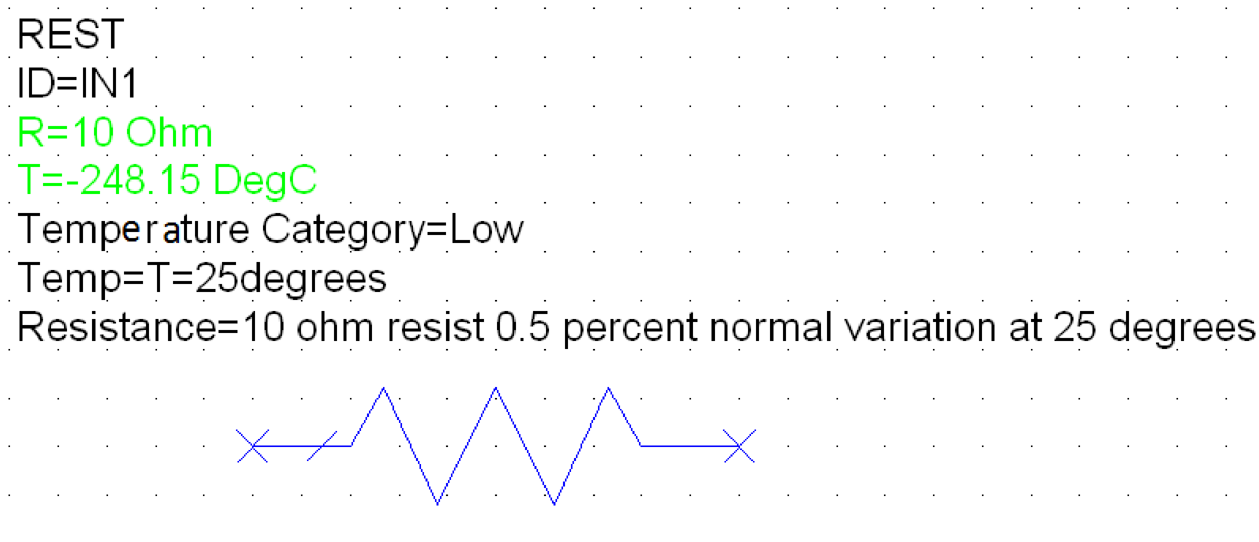
Figure A.28. Excel Spreadsheet Example for Parameterized XML for VSS

	A	B	C	D	E	F	G	H
1	Folder	xml model type	parameter name	parameter listing column	top param	ICON		
2	parameterized	awrmodel	Resistor	A		Filters		
3	COMPONENT	MODEL	DESC	PARTNUMBER	SYMBOL	HELP	CELL	Data
4	r1 SP	LIN_S	10 ohms	r1sp		Help\Helpsparam.doc		sparam\r10.s2p
5	r2 SP	LIN_S	20 ohms	r2sp		Help\Helpsparam.doc		sparam\r20.s2p
6	r3 SP	LIN_S	30 ohms	r3sp		Help\Helpsparam.doc		sparam\r30.s2p
7	End							

A.9.2.3. Parameterized XML Through Multiple Layers of Hierarchy

Another advantage of parameterization is that you can parameterize through multiple layers of hierarchy. Note that you can do this only if the components are the same model type (for example, S-parameters). Figure A.29, "Example of Multiple Layer of Parameterization" is an example of this application that creates two categories of low and high temperature resistors. In each category you can select from three different resistor values with different temperatures.

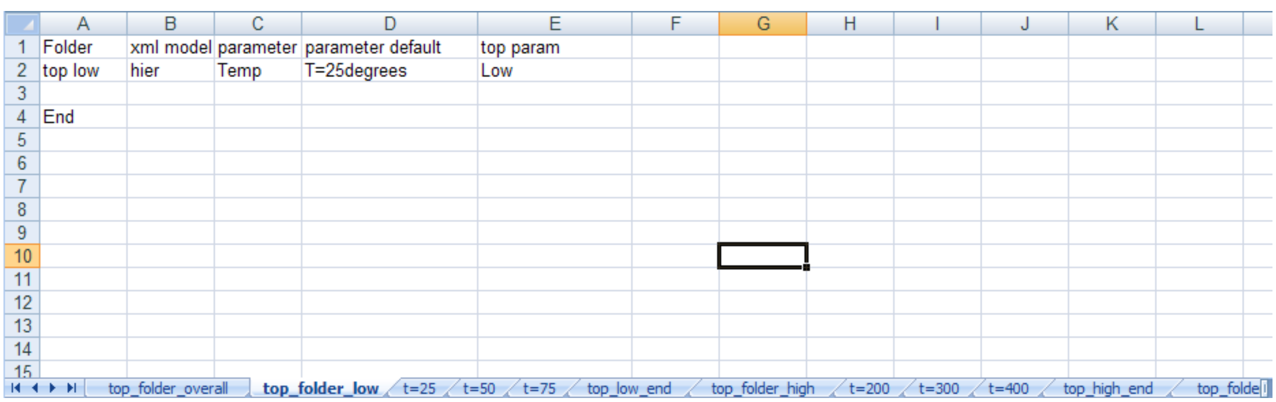
Figure A.29. Example of Multiple Layer of Parameterization



To create a hierarchy of parameterized folders, there are some things you need to do differently from the preceding section:

- You must add at least two more Excel spreadsheets (the number of additional spreadsheets depends on how many levels of hierarchy you want to create). Figure A.30, “Example of Excel Spreadsheet for Creating Multiple Levels of Hierarchy” shows the initial spreadsheet in which you start the hierarchy. Cell B2 must be "hier". In the second spreadsheet you end the hierarchy by filling cell B2 with "**hier_end**".
- In your lower levels of hierarchy, you must fill out cell E2 which is "top parameter value".

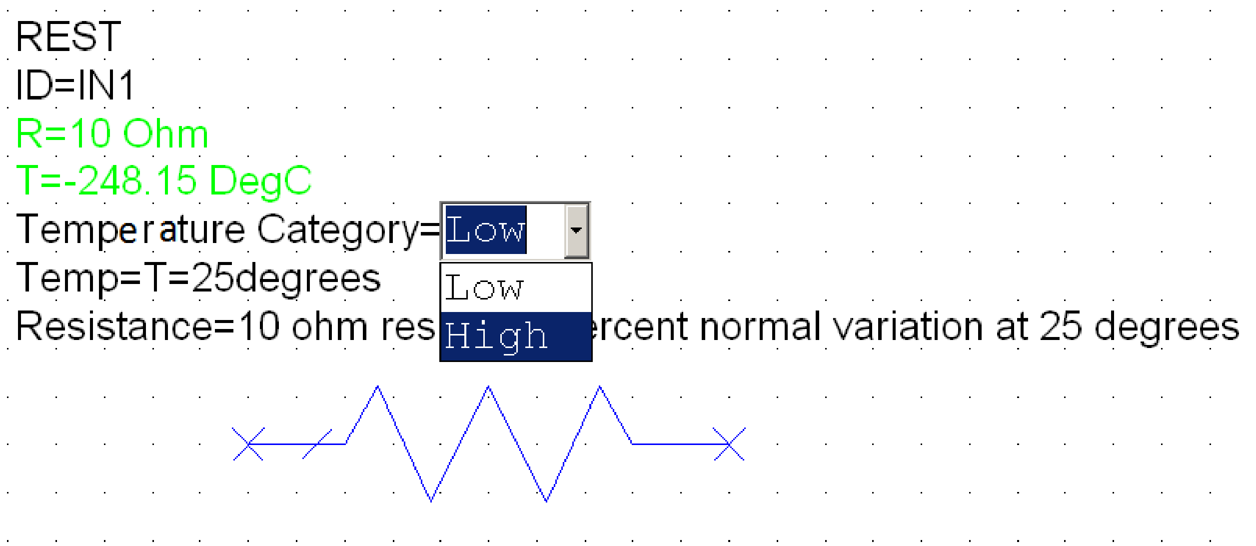
Figure A.30. Example of Excel Spreadsheet for Creating Multiple Levels of Hierarchy



	A	B	C	D	E
1	Folder	xml model	parameter	parameter default	top param
2	top low	hier	Temp	T=25degrees	Low
3					
4	End				

Figure A.31, “Example of Parameterized Element with Multiple Levels of Hierarchy”shows the finished element.

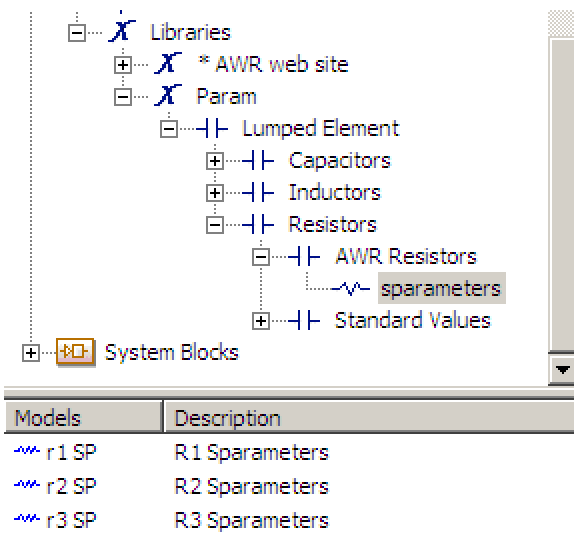
Figure A.31. Example of Parameterized Element with Multiple Levels of Hierarchy



A.9.3. Using Parameterized XML

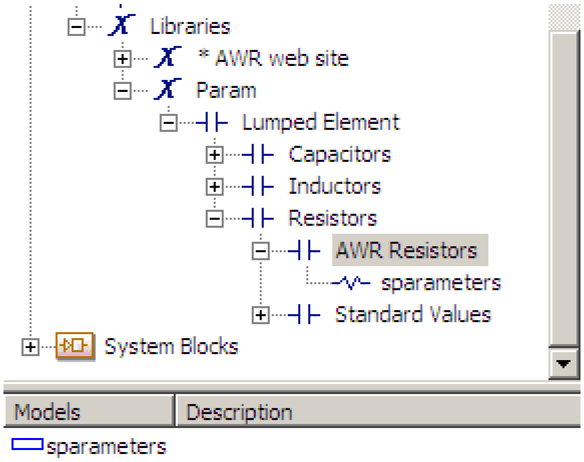
After you configure parameterized XML, you see new models available one folder level higher than where the individual models reside. In the previous example there are three resistors in the **sparameters** folder of the library, as shown in Figure A.32, "sparameters Folder Contents".

Figure A.32. sparameters Folder Contents



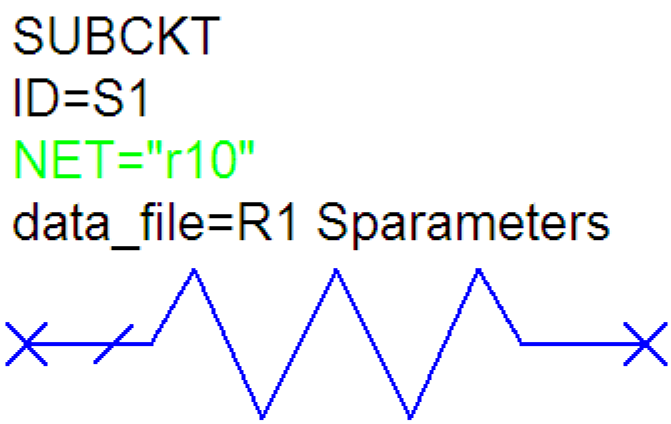
If you click one folder up, as shown in Figure A.33, "Parameterized XML Element in Elements Browser" you see one model in the lower window-- this is the parameterized XML element.

Figure A.33. Parameterized XML Element in Elements Browser



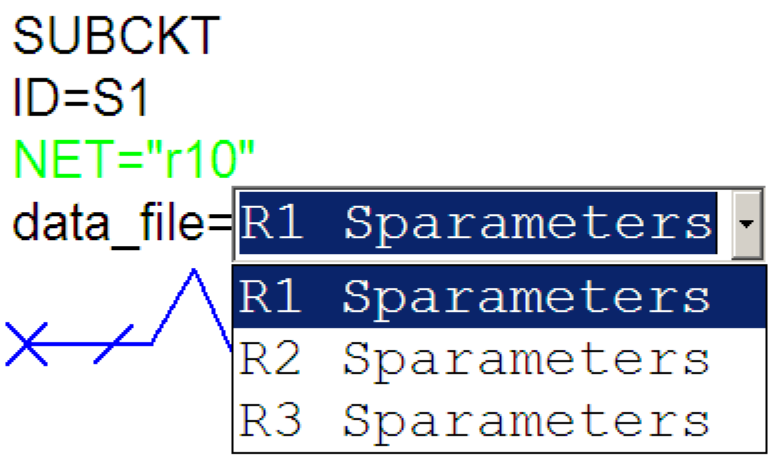
You click and drag this element into a schematic to create a parameterized XML model as shown in Figure A.34, "Parameterized Element Parameters". Note that the NET parameter is 'read only' and there is a new data_file parameter defined by the ParamName attribute for the FOLDER element.

Figure A.34. Parameterized Element Parameters



The values available for the data_file parameter are show in Figure A.35, "data_file Parameter Values". These match the values set for the Param attribute for each COMPONENT.

Figure A.35. data_file Parameter Values



A.9.4. Parameterized XML Limitations

Parameterized XML is only supported for awrmodel, sparameters, and awrschematic types.

When you create parameterized XML, all of the components must be the same model type. For example, you cannot mix between awrmodel and sparameters model types.

When you create parameterized XML that uses an external file (such as sparameters or awrschematic):

- Each time you switch to a new part, a new file is loaded into the AWR Design Environment platform. If you are optimizing or tuning this may result in many files loaded into your project.
- The model definition must match the SUBCKT. If you are not using parameterization, you can leave the model element blank and the library still functions properly.

A.9.5. Parameterized Subcircuits

A parameterized subcircuit is created to allow you to easily define custom models within the schematic capture interface. It can also allow subcircuits to use values passed in from schematics at a higher level in the hierarchy. The XML format allows you to define the symbol, layout cell, description, and Help file of a new element. The Excel spreadsheet, the schematic, and the XML file used in this appendix are available via the "XML Library Development Application Note" in the Cadence AWR Knowledge Base. You can locate this application note and associated files by searching for the term "library development" in the Knowledge Base. For more information on using parameterized subcircuits with layout, see “Using Parameterized Subcircuits with Layout”.

See “Using the AppDataUser Folders” for information on how to call this library in the AWR Design Environment platform.

A.9.5.1. Parameterized Subcircuit Example

This example demonstrates how to create and use a parameterized subcircuit. Figure A.36, “Example of Parameterized Schematic” shows a schematic with parameterized network:

Figure A.36. Example of Parameterized Schematic

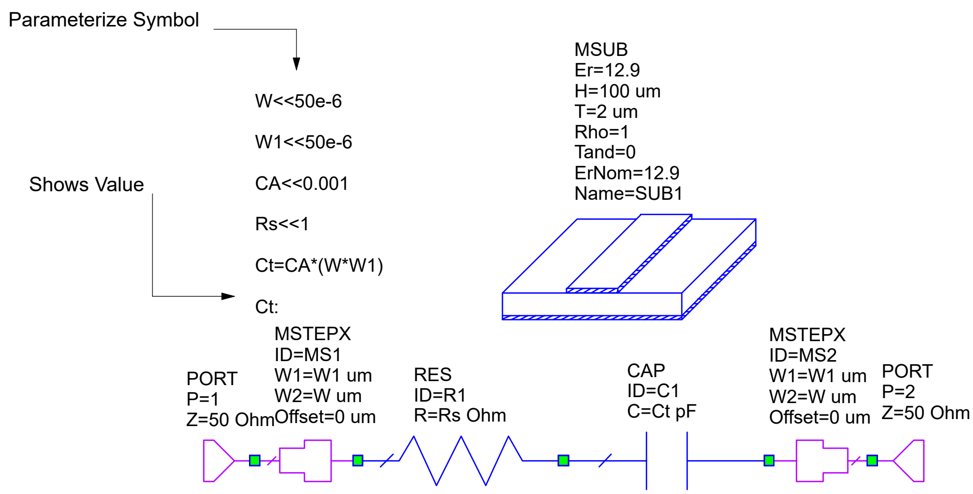
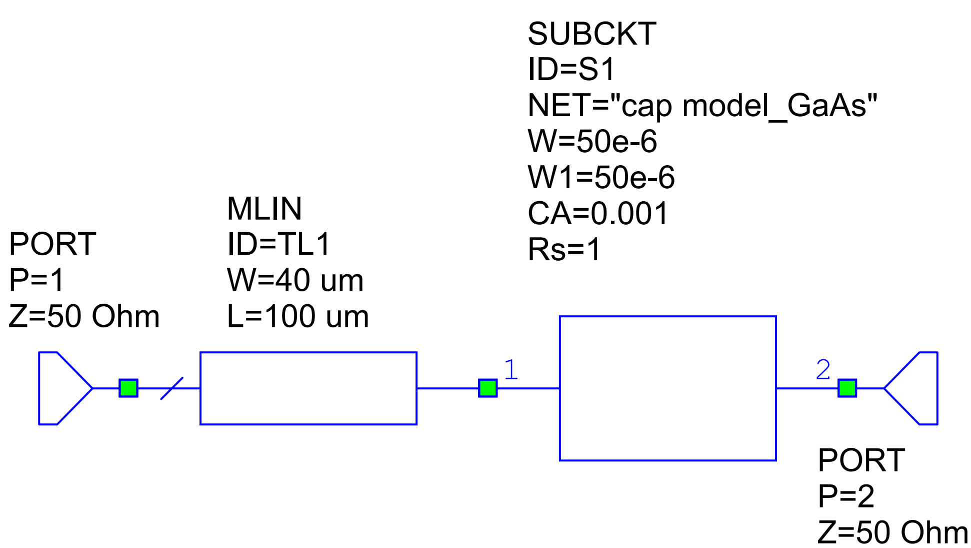


Figure A.37, "Utilization of Parameterized Subcircuit in Another Schematic" shows how the parameterized subcircuit in Figure A.36, "Example of Parameterized Schematic" can be used in another schematic. Notice that the parameterized values such as W, W1, CA, and Rs got passed to this schematic. You can change these values by editing the original subcircuit values. The passed-in variables do not have units, but when they are assigned to an element that has units, they use the element units.

Figure A.37. Utilization of Parameterized Subcircuit in Another Schematic



A.9.5.2. Creating Parameterized Subcircuits

Creating a parameterized subcircuit involves creating variables and equations to express parameter values. The following is an example of creating the Excel spreadsheet for a parameterized subcircuit. Figure A.38, "Excel Spreadsheet Example for Parameterized Subcircuit" is the Excel spreadsheet for Figure A.36, "Example of Parameterized Schematic". You need to specify SUBCKT as the model and awrschematic as the xml model type. Similar to the AWR model, you can have 'read-only' values as well as different distribution for the values as shown in Figure A.38, "Excel Spreadsheet Example for Parameterized Subcircuit":

Figure A.38. Excel Spreadsheet Example for Parameterized Subcircuit

A	B	C	D	E	F	G	H	I	J	K	L
Folder	xml model type	parameter name	parameter listing column		ICON						
	awrschematic				Capacitor						
COMPONENT	MODEL	DESC	PARTNUMBER	SYMBOL	HELP	CELL	DATA	W	W1*	CA	Rs
MIM Cap GaAs	SUBCKT			Capacitor@system.syf			cap model_GaAs.sch	5.0E-05	5.00E-05	0.001	1 s(1p,u)
End											

A.9.6. Generating MDIF Files

For Microwave Office libraries, you may want to use MDIF files for library parts. An MDIF file is a collection of 2-port S-parameters in one file that has a parameter setting to select which S-parameter block is used in simulation. For AWR XML libraries, you should only use MDIF files for a single part that is measured with different configurations. A good example is a device measured at different bias points, or one part measured at different locations across a wafer, because there is only one part number to the MDIF file. Consequently, the MDIF should only contain data on one unique part.

A.10. Troubleshooting

There are several steps you can take to help debug your XML library while creating it.

A.10.1. Debugging XML Files

When generating XML files, errors are often encountered. Understanding how and when XML files are read dramatically speeds up debugging. When a project is open and an XML library folder is selected, the entire XML library is read. The program does not reread the XML file when subsequent parts are selected from that library, therefore, it is not possible to change an XML file and see the changes in an open project. To re-read the XML file, you must open a new project. You can choose **File > New Project** to open a new file rather than shutting down and restarting the software.

A.10.2. Validating the XML

The AWR Design Environment software validates your XML as you navigate through your library. When you click on a folder and a new XML file is read, the XML is validated and any errors are listed in the Status Window.

For example, the following XML has an error in the PARAM element:

```
<DATA DataType="sparameter">
 sparam/r10.s2p
 <PARAM Name="NET" ReadOnly="yes"></PARAM>
</DATA>
```

When Cadence AWR reads this XML, the Status Window displays an error:

```
11:40:45 AM   Element content is invalid according to the DTD/Schema.
Expecting: #PCDATA, {urn:awr-lib-data}PARAM, {urn:awr-lib-data}VARIABLES,
{urn:awr-lib-data}PARAMSDATA, {urn:awr-lib-data}SDATA,
{urn:a....    Error on line 154, position 37 in
"file:///C:/8_0_xml/test/AWR_Resistors_Lib/AWR_Resistors.xml".
```

A.10.3. XML Verbose Mode

You can instruct the AWR Design Environment platform to show you more XML reader details if you need help debugging. In your *user.ini* file, add the following text:

```
[XmlLibs]
VerboseLog=1
```

After you restart the AWR Design Environment platform, each XML file that is read reports information in the Status Window. For example, when you first start the program you see:

```
XML Libraries
 Loading schema file://C:\Program Files\AWR\AWR2022_3723_2\library\libschema.xml
 Loading XML library file://C:\Program Files\AWR\AWR2022_3723_2\library\lib.xml
 Loading XML library file://C:\Program Files\AWR\AWR2022_3723_2\library\sys_lib.xml
```

NOTE: Your program directory may be `//C:\Program Files (x86)`. Each file is a link so you can click it to view the file contents.

A.10.4. Testing the XML Library Using a Visual Basic Script

There is an AWR Design Environment platform project named *Test XML Libraries.emp* that is available via the "XML Library Development Application Note" in the Cadence AWR Knowledge Base. You can locate this application note and associated files by searching for the term "library development" in the Knowledge Base. Loaded in this project is a Visual Basic script named *autotest_specific_xml* that tests for errors any XML library that you specify within the code. The script generates two log files: *XML_Lib_specific.log* and *XML_Lib_specific_error.log* in the directory where the *Test Xml Libraries.emp* project is located. The first one lists all of the levels of the libraries under test, including the error messages after each part that fails. The second one contains only the error messages for each part.

To specify the library on which you want to perform the test, you need to modify the following line of code in the script:

```
Set libs = MWOffice.Application.Libraries("Circuit Elements").
                Libraries("Libraries").Libraries(1).Libraries(
                2).Libraries
```

This line of code performs the test on the first node under **Libraries** in the Elements Browser. Under the ***AWR web site** library this is **Parts By Type**. For individual use, if the library that you test is the third node under **Libraries**, you need to modify the above line of code as follows:

```
Set libs = MWOffice.Application.Libraries("Circuit Elements").
                Libraries("Libraries").Libraries(3).Libraries
```

See "The Script Development Environment" for more information on working with the scripting environment.

NOTE: As of version 9.0, this test script is not functional for VSS libraries.

Appendix B. New Design Considerations

This appendix provides guidelines for starting a new design. The information included here is based upon years of experience supporting Cadence® AWR Design Environment® platform users with problems that can arise late in a design cycle. This information is intended to help designers make decisions at the beginning of the design cycle to avoid problems later.

B.1. Overview of Considerations for a New Design

The following design issues are discussed in detail:

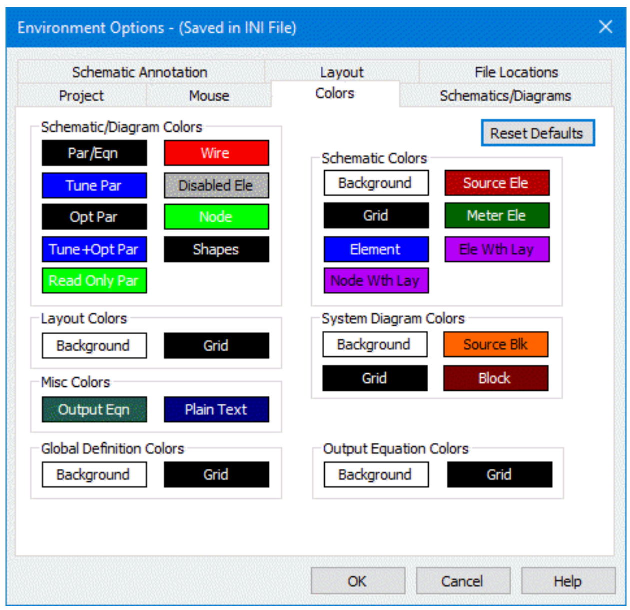
- User Interface Issues
  - Design Environment Colors: Set up the AWR Design Environment platform with your preferred schematic, system diagram, and layout colors.
- Simulation Issues
  - Project Units: Determine your project units strategy-- either setting unit modifiers or only using base units. Project units are problematic when using equations if you change your project units, or if you export a schematic from one project and import it into a project with different units.
  - Dependent Parameters: Standardize the units to Base Units prior to exporting to ensure that a consistent set of schematic parameter values are used for group design projects.
  - Test Benches: A hierarchical approach to simulation that requires creation of a single schematic with all circuit elements, except sources. After this step, a test bench is created with the appropriate sources, using a copy of the original schematic for specific measurements. The advantage of this approach is that the original schematic is not altered in any way.
  - Multi-processor Configuration: Take advantage of multiple processors in a machine for electromagnetic and harmonic balance simulations.
  - X-Models: X-models, or EM-based models, are used for distributed elements when the highest degree of accuracy in representing a discontinuity, such as a bend or tee, is desired. When properly used, X-models offer the accuracy of EM simulation with the speed of linear simulation.
- Layout Issues
  - Database Resolution: A sensible database resolution unit is important when setting up a process. This step is automated for designers using Process Development Kits (PDKs). A database unit is defined as the smallest unit of precision for a layout.
  - Flow Considerations: Working through a planned design flow is recommended before beginning an actual design, to eliminate potential problems late in the cycle. The manufacturing flow, layer configuration, artwork import, and design export must all be considered before starting the design.
  - Snapping Mode: Snapping functions connect the faces of layout cells in different ways. You can set snapping options in the Layout Options dialog box. Cadence recommends using manual snapping rather than automatic snapping for complex layouts.
  - Inset Faces: In PCB, MIC, or MMIC designs, connections between cells are made on faces rather than pins. Gaps can be created in layout as a result of rotating cells if layout options are not fully understood.
  - Export Options: It is important to understand Layout Options if you plan to export your layout. You do not need to select these options at the beginning of a project, however, you should consider them prior to tape-out.
  - GDSII Cell Options: It is important to understand the implications of selecting the various Cell Library properties settings.

  - Physical Considerations: If you plan to perform Layout Versus Schematic (LVS) analysis on Cadence Microwave Office® software projects, you should be aware that an LVS engine may not understand all portions of your design, such as EM structures or Touchstone® data files.
- Library Issues
  - Component Libraries: Decide on a vendor library management approach.

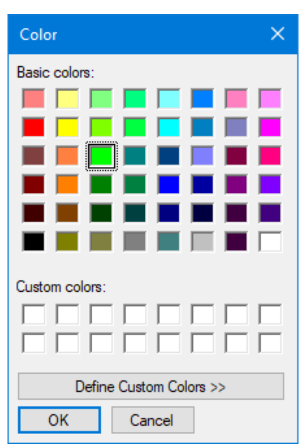
B.2. Configuring Schematic and Layout Colors

It is simple to configure the colors used to display various objects on schematics, system diagrams, and layouts. Designers who use certain colors throughout their design careers find that the AWR Design Environment platform makes it easy to set the desired colors either prior to, or during the design process.

To change environment colors, choose **Options > Environment Options** and click the **Colors** tab.



To change the color of an object, click on the name of the object to display a Color dialog box from which you can select a color.



To return to the original settings, you can click the **Reset Defaults** button on the **Colors** tab. Settings are saved for each user, and used every time you run the AWR Design Environment software from the same computer.

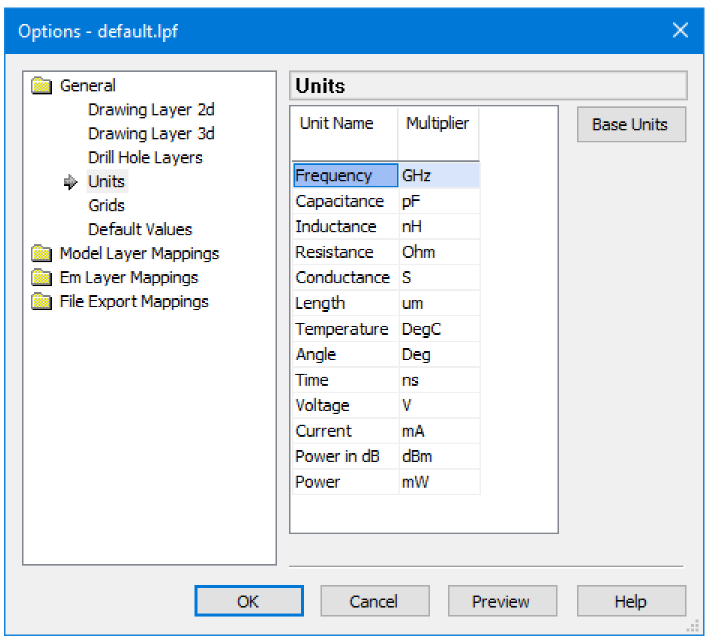
The AWR Design Environment platform also includes fully customizable menus, toolbars, and hotkeys. See "Customizing the Design Environment" for information on customization.

B.3. Determining Project Units

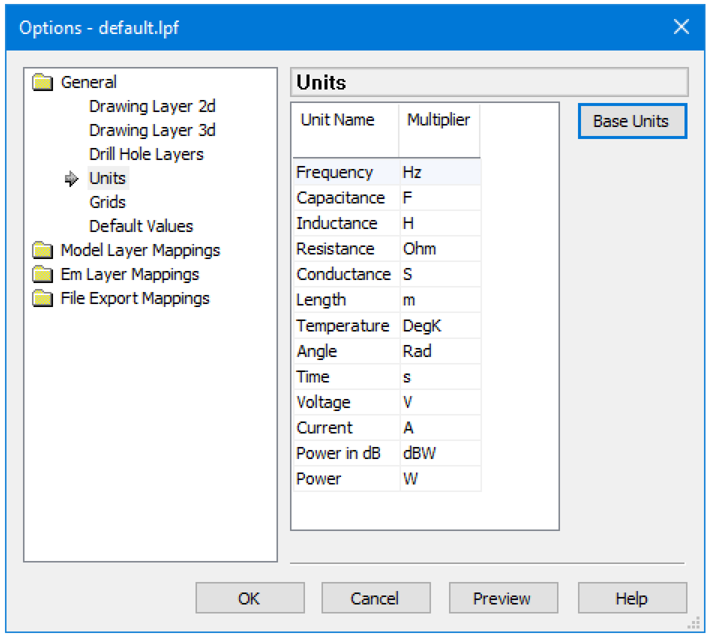
Before designing in the AWR Design Environment platform, you should have a sound strategy for dealing with project units. AWR Design Environment software now supports per-process technology native units, allowing different designs within the same project to specify values in different units. Units used for each process are specified as part of a Layer Process File (LPF). For more information on working with LPF files, see "The Layout Process File (LPF)".

B.3.1. Configuring Units with Layout

When using the AWR Design Environment software with the Layout feature license, you configure LPF units by choosing **Options > Drawing Layers**, then selecting the desired LPF from the Select LPF file dialog box. Click **OK** to display the LPF Options dialog box, then under the **General** folder in the left pane, click **Units** and specify the desired units.

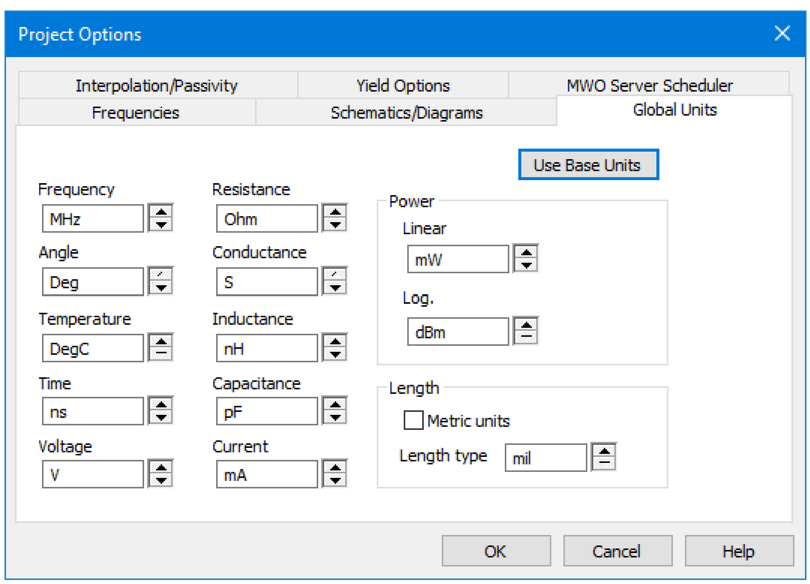


You can set project units (for instance, pF for capacitance) as shown in the previous figure, or you can work in base units (Farads for capacitance), as shown in the following figure. To set all modifiers to base units, click the **Base Units** button. If using base units, you can use unit modifiers in your schematics to convert the units.

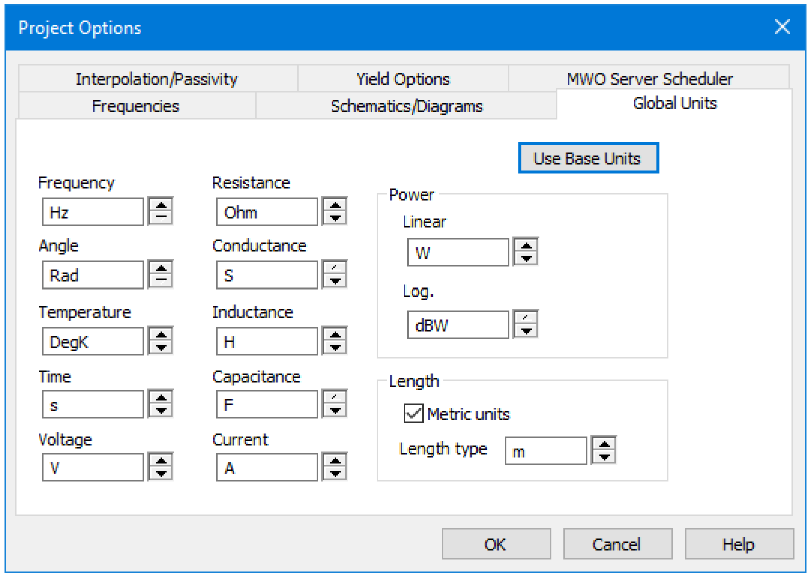


B.3.2. Configuring Project Units without Layout

When using AWR Design Environment software without the Layout feature license, units are configured in the Project Options dialog box by choosing **Options > Project Options** to display the Project Options dialog box, then clicking the **Global Units** tab.



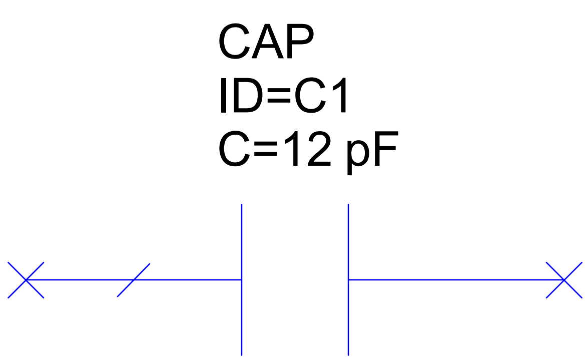
You can set project units (for instance, pF for capacitance) as shown in the previous figure, or you can work in base units (Farads for capacitance), as shown in the following figure. To set all modifiers to base units, click the **Use Base Units** button. If using base units, you can use unit modifiers in your schematics to convert the units.



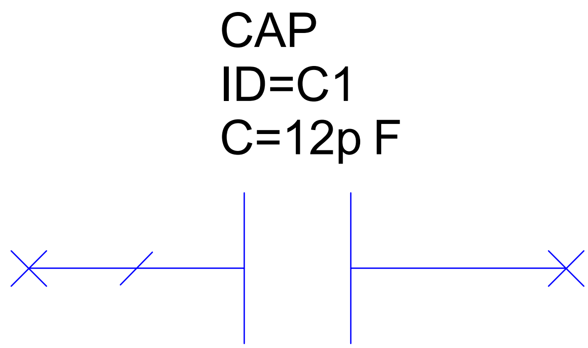
The following is a list of standard (non-case sensitive) unit modifiers:

f	1e-15
p	1e-12
n	1e-9
u	1e-6
m	1e-3
c	1e-2
mil	25.4e-6
k	1e3
meg	1e6
g	1e9
t	1e12

For example, if using project units as pico for capacitance, you could simply enter "**12**" as the value of the schematic element CAP to represent 12 pF.



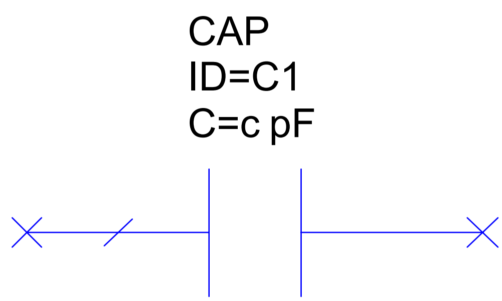
If using base units, you need to enter "**1.2e-11**" or "**12p**" (if using a unit modifier) as the value of the schematic element CAP to represent 12 pF.



When you use base units, values display the same way whether typed with or without a unit modifier.

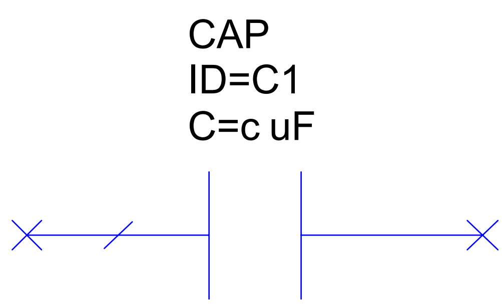
It is important to remember which approach you take if you plan to use equations in your project, because equations do not have units. For example, suppose you set capacitance units to pF, as in the previous example. Next, you create a variable and assign it to the value of the capacitor.

c=12



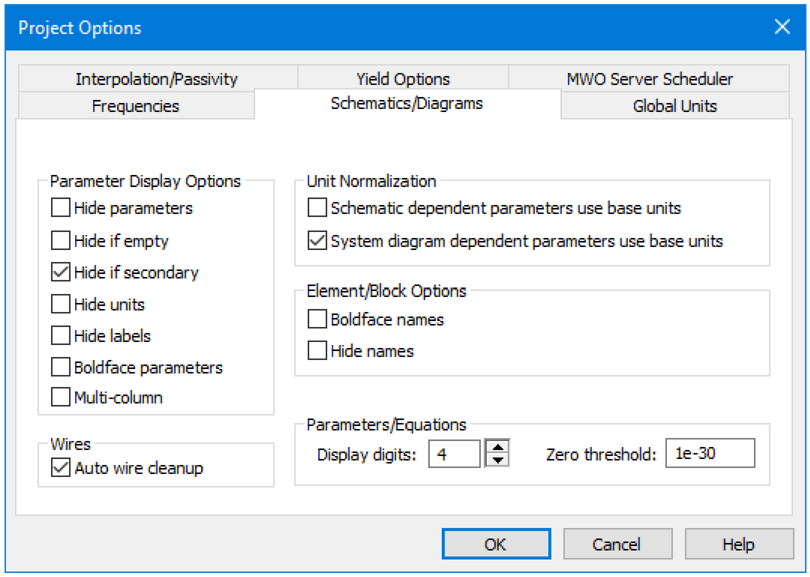
If you then change the global (project) units in the Project Options dialog box from pF to uF for capacitors, the variable does not scale-- no conversion factor is applied to your equation to account for the change in units.

c=12

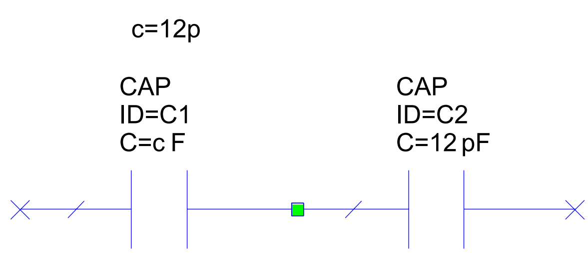


This change can produce a result that is orders of magnitude from the expected answer.

Typically, units are not changed after a design is started. It is possible, however, for one designer to export a schematic using one set of units, and for a second designer to import the same schematic using a different set of units. It is especially difficult if the project includes variables and equations. The AWR Design Environment platform includes an option that can help. Choose **Options > Project Options** and click the **Schematics/Diagrams** tab.



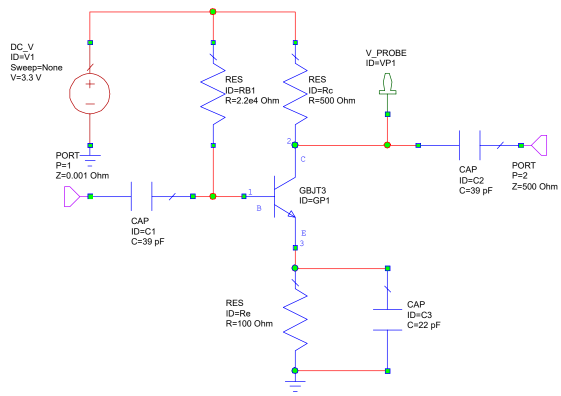
Select the appropriate **Unit Normalization** check box to specify that any parameter assigned to a variable always uses base units, as shown in the following example.



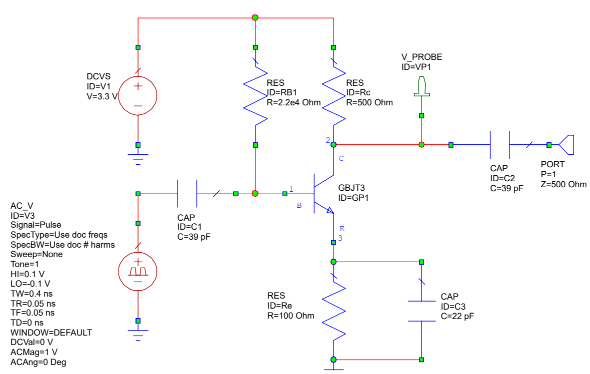
The disadvantage of this mode of operation is that the designer must remember to enter each variable value in base unit values. The advantage is that there are never unit scaling issues if project units change, or when schematics are shared between designers. Cadence strongly recommends using this setting when exporting and importing between projects to eliminate a unit scaling problem.

B.4. Using Test Bench to Analyze Designs

This section provides information for users who need to perform many different types of analysis on the same schematic. The "flat" approach, as it is commonly called, involves using duplicated versions of a schematic for various types of measurements. This is done by copying and pasting an original schematic into a new window and making minor modifications (such as changing a source or a port) specific to a particular measurement. For example, to measure S11 of a circuit and also input a voltage square wave, with the flat approach you have one schematic for S11

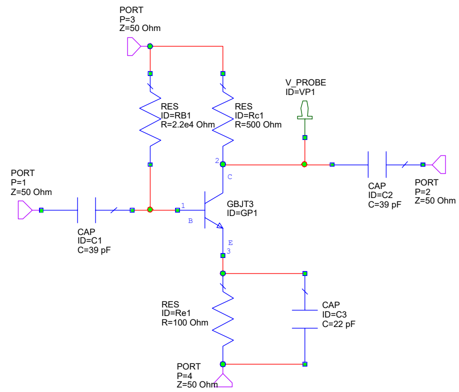


and one schematic for your voltage square wave.

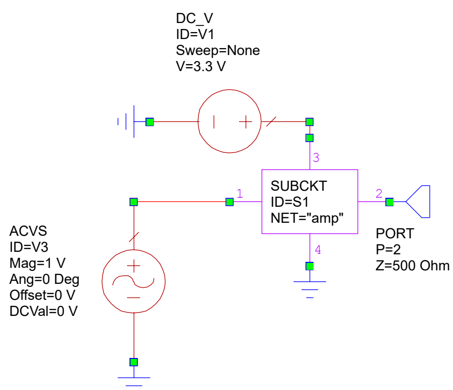


This approach is very error prone and time consuming, as a designer must ensure that all of the flat versions of the design are the same. This method becomes more difficult as additional types of analysis are required.

A more organized approach, known as the "Test bench", requires that a design be created with all of the elements included (for example, microstrips, passive, and active) except sources. In this example, the new design appears as follows:



A test bench is created for each specific measurement to be performed, and it includes the design as a subcircuit with the sources and terminations added.



The advantage to this approach is that the original schematic is not altered, thus keeping the design process more organized.

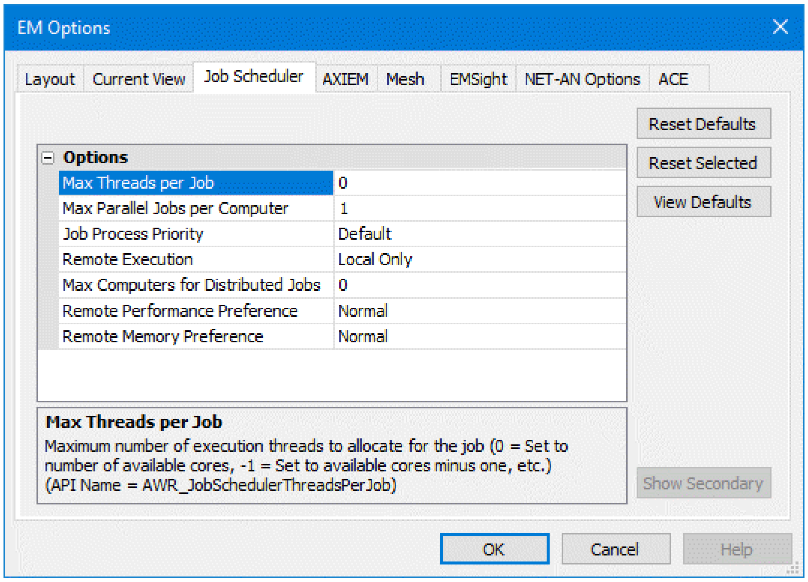
Designers are commonly confused regarding how the ports at the design level load their circuit. Remember that in a simulation, only the top level port impedances are considered. Ports in a subcircuit are used only to define connectivity.

Note that the subcircuit symbol can either be user-defined or created using the Symbol Generator Wizard. See *"Circuit Symbols"* for information about creating custom symbols for your subcircuit.

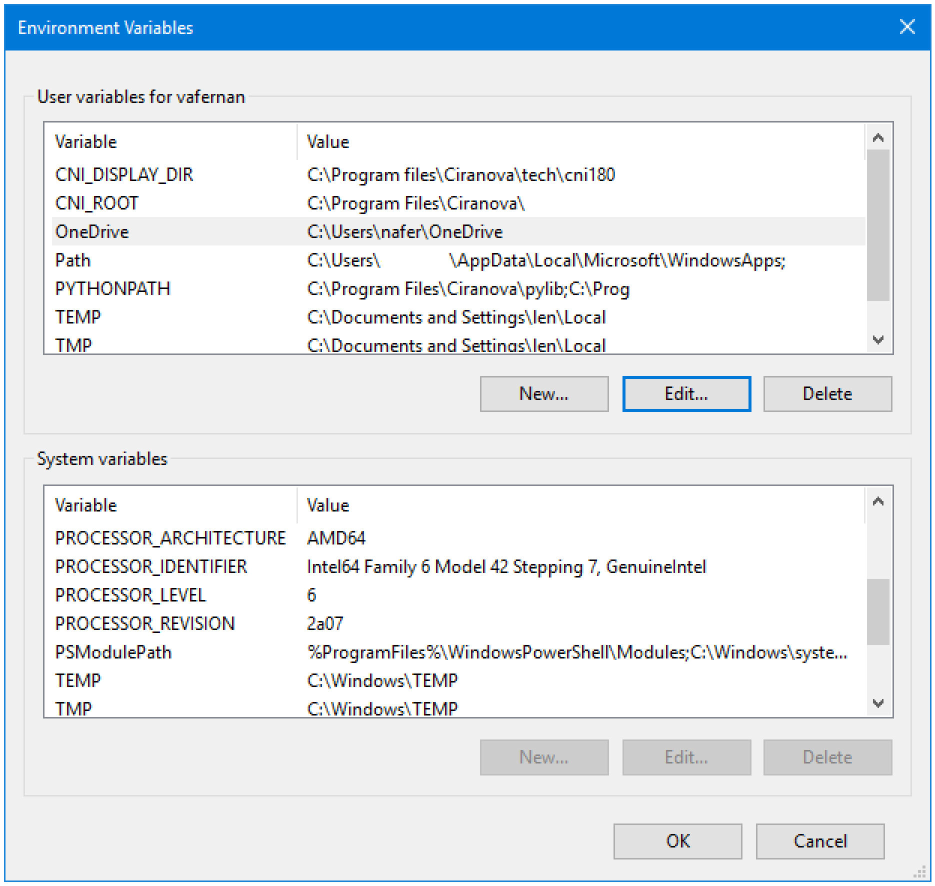
B.5. Multiple Processor Setup

You can configure the Cadence AXIEM® 3D planar EM simulator, Cadence Analyst™ 3D FEM EM simulator, and EMSight simulator to use multiple processors. Each simulator utilizes a different method for setting the number of processors for it to use.

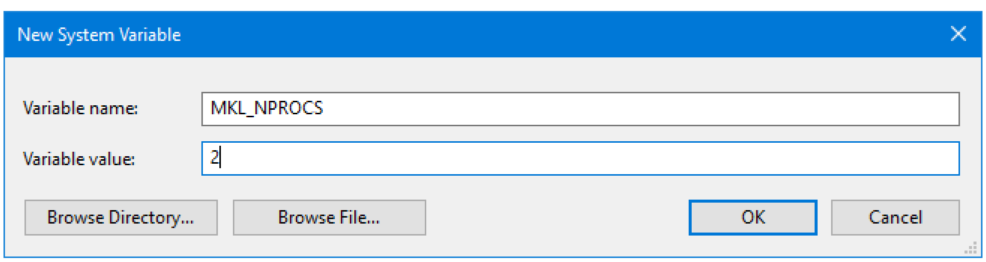
To set the number of processors for AXIEM or Analyst software simulators, you specify a value for **Max Threads per Job** on the **Job Scheduler** tab. If set to "0", the maximum number of processors, up to eight, are used. To specify this setting for individual EM documents, clear the **Use project defaults** check box on the Options dialog box **Job Scheduler** tab for the specific EM document. To specify this setting globally do so on the EM Options dialog box **Job Scheduler** tab.



To set the number of processors for EMSight, you need to configure two system environment variables. To access these variables, in the Windows® Control Panel in the "Find a setting" search box, type "Environment Variables". Choose the system environment variables option when prompted. In the System Properties dialog box that displays, click on the **Environment Variables** button at the bottom.



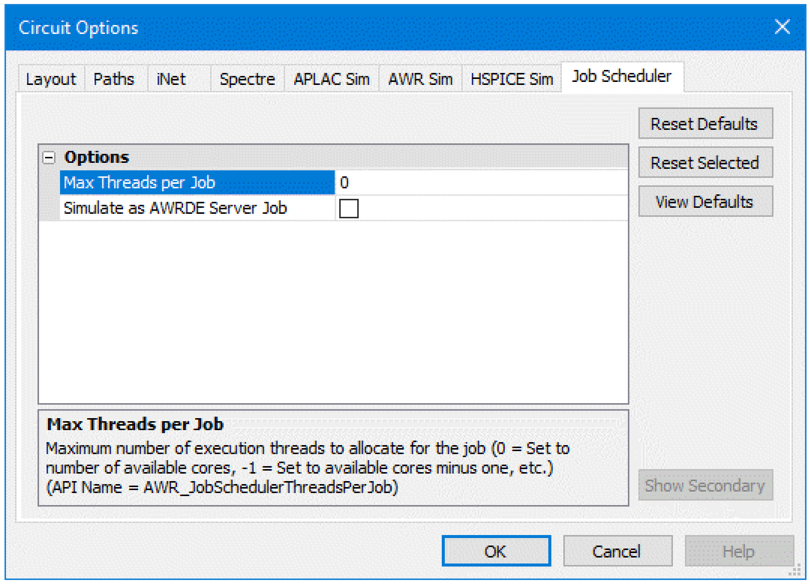
Click the **New** button below the **System Variables** section to open the New System Variable dialog box. Add new "MKL_NPROCS" and "OMP_NUM_THREADS" settings, as shown in the following figure.



If using two processors, set the **Variable value** to "2", and likewise for other numbers of processors.

Note that after setting **Max Threads** to a value greater than 1, you do not see 100% use of the processor during the entire simulation run. There are three phases of an EM simulation: creating the Green's functions, filling the solution matrix, and solving the solution matrix. It is only during the solving of the solution matrix that multiple processors might be used (if enabled). The percentage of time spent "solving" as a function of total simulation time varies depending on the problem.

A **Max Threads** option on the Circuit Options dialog box **Job Scheduler** tab allows you to set this option globally for the project. You can also set it per document in the Options dialog box (right-click the schematic in the Project Browser and choose **Options**).



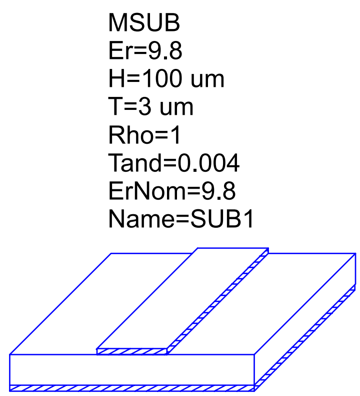
B.6. Using X-models

X-models, or EM-based models, are used for distributed elements when the highest degree of accuracy in representing a discontinuity, such as a bend or tee, is desired. X-models use a table lookup approach to access the results of EM simulations and then interpolate to specific data points. When all points are filled in the database and saved in a file, the simulation is extremely fast with EM-based accuracy.

A common mistake is attempting to use an X-model in a design that does not have the tables filled. A designer generally attempts to solve this problem by setting the Autofill parameter of that model to "1" to fill the table. The simulations to fill the X-model tables for all discontinuity models can take many hours to complete. It may not be necessary to fill a table, however, as many pre-filled tables are shipped with the AWR Design Environment software. If the nominal dielectric values of the pre-filled tables are within 10% of the values you need, no filling is required.

If you decide to use X-models in your design, you need to decide whether or not you need to fill an X-model database. AWR software is shipped with many common X-model tables already prepared.

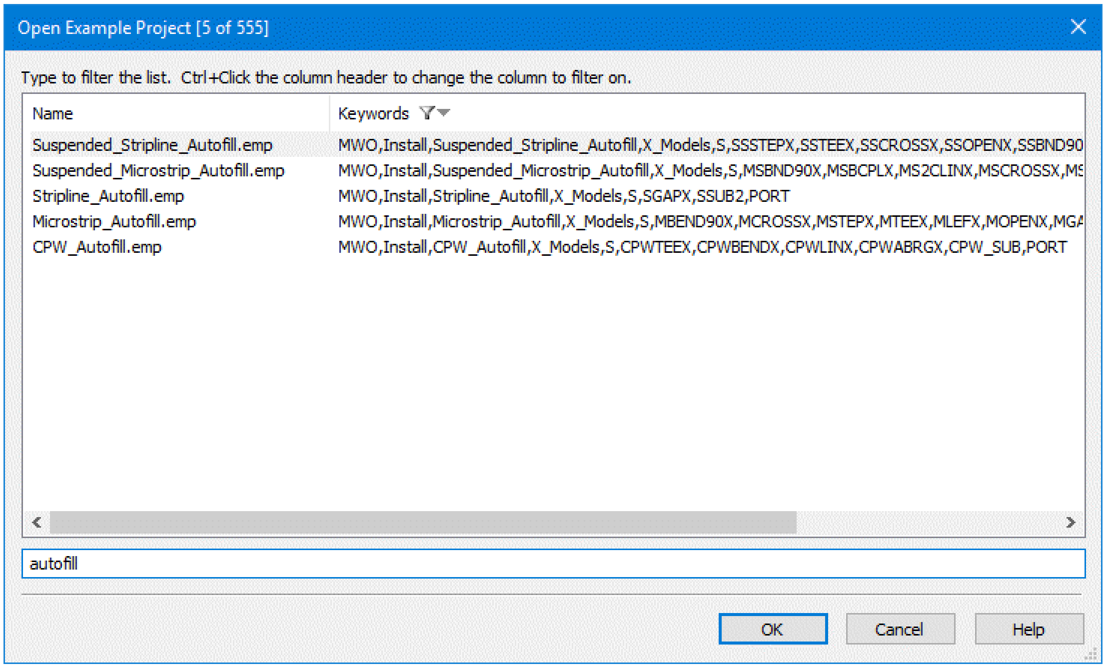
To determine if tables are already filled, you must understand the different types of parameters used for X-models. For each substrate that has X-models, there are one or more "nominal" parameter values specified that are used for EM simulation. For example, consider the following MSUB substrate.



This element has both an "Er" and "ErNom" parameter. The "nominal" parameter is used to look up the X-model table. The non-"nominal" value is used in simulation. The non-"nominal" values can vary from the "nominal" values by 10% before the simulator issues a warning. This allows you to use pre-filled X-model tables for statistical analysis.

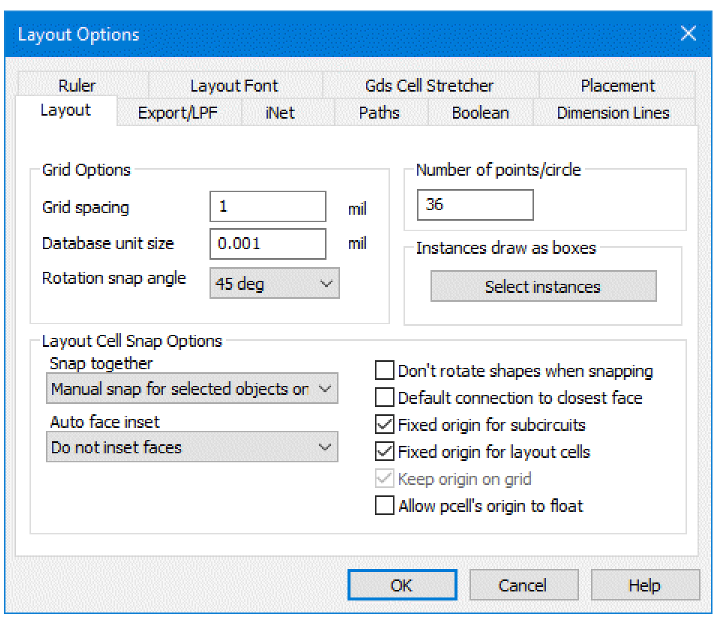
When you know what your "nominal" parameter values are, you can search in the folder that contains the X-models shipped with the software to see if a table exists. To see which tables are filled, choose **Help > Show Files/Directories**. Double-click the *application* folder to open your installation directory, and then double-click the *EM_Models* folder to view the various model files. Note that the relevant parameters corresponding to each model are included in the file name. Remember that the non-"nom" values can vary by 10% from the "nom" value, so if your exact value isn't there, you can use one close to your value. For example, if you are using microstrip and your Er is 10, you will find that tables for 9.8 and 10.2 are already filled. Therefore, you can set "ErNom" to 10.2 or 9.8 and set "Er" to 10 and it works.

Filling a table is a time-consuming process but it only has to be done once. If you decide you need to fill your own tables, Cadence highly recommends using the projects shipped with the software to help you do so. Currently there are projects configured for all the substrates that have X-models. To find these projects, choose **Help > Open Example** to display the Open Example Project dialog box. Type "**autofill**" at the bottom of the dialog box to filter just the X-model autofill projects. Select the project you want and make sure to read the Design Notes before beginning.



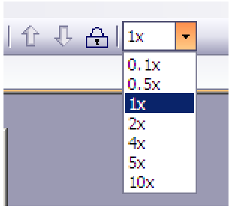
B.7. Determining your Database Resolution

A sensible database resolution unit is important for setting up a process. This step is automated for designers using Process Development Kits (PDKs). A database unit is defined as the smallest unit of precision for a layout. All layout coordinates in the AWR Design Environment platform are stored with 1nm resolution, however, most processes cannot resolve 1nm geometry so the database resolution allows you to match the layout precision with your process precision. To set the database resolution for a process, choose **Options > Layout Options** to display the Layout Options dialog box, then click the **Layout** tab. Under **Grid Options**, select an LPF from **Grid to modify** and then click the **Edit Grid** button to display the LPF Options dialog box where you can specify the **Unit** for the Database Size.



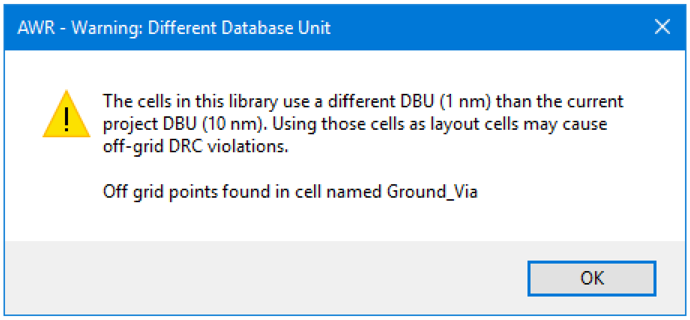
The minimum value when working in metric units is 1nm, and when working in English units is 0.005 mils. Typically, values are determined by the foundry for integrated circuits. If your process has no vertex limitations, Cadence strongly recommends using the minimum dbu. Cadence also recommends that you set this at the beginning of your design and do not change it after you have begun. If you must change it, you should inspect your entire layout for any gaps or shorts.

Grid spacing specifies the drawing grid snap, which simplifies the placement and construction of shapes within the layout system. On the Schematic Layout toolbar, the **Grid Spacing** button includes a drop-down menu that allows you to select a multiplier for the current grid.



Because the grid multiplier's smallest unit is 0.1x, you should set the grid to at least 10 times the database unit.

Setting the database unit can be confusing when GDSII libraries are imported into the AWR Design Environment platform. You can import these libraries in the Artwork Cell Editor or when bringing in a vendor library part. Each GDSII file has its own database resolution, and if the file resolution doesn't match the project resolution, a warning similar to the following displays:



Do not be alarmed by this message as the cells in the GDSII library are rounded to the database resolution you specify, so you are not losing any design accuracy relative to your database resolution.

B.8. Using Dependent Parameters

This option tells any model that has a variable assigned to use base units (for example, amps, volts, or meters). You should use this option when exporting and importing schematics between projects, because different users may have different Global Units. One designer may use pF for capacitance and another may use uF. If they share projects, they will likely not import with the intended values in their schematics. Standardizing the units to base units prior to exporting ensures that a consistent set of schematic parameter values are used by both designers.

There are separate controls for schematics and system diagrams. For more information about this option see “Project Options Dialog Box: Schematics/Diagrams Tab ”.

B.9. Configuration for PCB Layout and Manufacturing

The AWR Design Environment software can export layout in GDSII, DXF, and Gerber formats. Typically, integrated circuit designers configure the AWR Design Environment platform for their process when they use a PDK to design their circuits, and can export their designs from the AWR Design Environment platform as GDSII layout files to successfully transfer their designs for manufacturing. PCB or MIC designers usually do not have the benefit of using a PDK to design, and the files needed for manufacturing are not as easily obtained as exporting a file from the AWR Design Environment software. You can configure the AWR Design Environment platform for a PCB process at any point during your design step. Cadence strongly recommends you work through your entire design flow before starting your design to avoid problems (which can lead to costly errors) at the very end of the design cycle. The following information includes topics to consider and references for further information.

If you plan to employ a PCB design tool such as Mentor Graphics Expedition or Cadence Allegro, contact Cadence Support to discuss integration options with these tools.

B.9.1. Manufacturing Flow

The AWR Design Environment platform was not designed as a complete PCB design and manufacturing tool. The value the AWR Design Environment software provides is accurately modeling RF structures and metal traces. The AWR Design Environment software does not directly support manufacturing data formats such as "Pick and Place", "Reference Designators", and "Bill of Materials".

B.9.2. Layer Configuration

Setting up the layout correctly for any process can be tricky for new users since some properties can be set in the AWR Design Environment platform itself and some must be set in the LPF file and then imported into your project. AWR Microwave Office Layout Guide covers each of these topics, however you should work through several application notes available in the Cadence AWR Knowledge Base:

- Balanced Amplifier Layout Application Note - Detailed Layout Application Note. This application note takes you from a blank process and goes through configuring your project for your design. It is rather lengthy, but you can work through just the beginning, which provides information on how to set up your process.
- Printed Circuit Board (PCB) Layout Application Note. This application note introduces the concept of positive and negative layers for helping you easily generate ground floods, and discusses various options for exporting Gerber with ground floods.
- Printed Circuit Board (PCB) Layout Application Note 2 - Multi-Level Layout. This application note shows how to set up a multi-level PCB layout. It provides information on how to make lines draw on the desired level in the board, and how to draw vias properly, since there are many different via types in this type of design. Finally, it demonstrates a new way of exporting NC drill file information for vias.
- If you plan to use ACE for a PCB design, a script exists to help you create the Layout Process File (LPF) for use with PCB iNets. To locate the "Create_PCB_iNet_LPF" script, search for "Created LPF" in the Cadence AWR Knowledge Base to find the associated article.

You can also set up layout using the Process Definition Wizard. In this case, the LPF file is defined as part of a Proccess Design Kit (PDK). See “Create New Process Wizard” for details.

B.9.3. Artwork Import

See “PCB Import Wizard” for details on importing 3Di, ODB++, and IPC-2581 standard files into the AWR Design Environment software. If you want to import DXF files, you should understand all of the DXF constructs supported in the Cadence DXF importer. See the Cadence AWR Knowledge Base "Using DXF Format with the AWR Design

Environment" application note on this topic. The AWR Design Environment platform cannot currently import Gerber files. If you need to import Gerber, Cadence recommends a file translation utility that you can purchase from Artwork Conversion Software, Inc..

B.9.4. Design Export

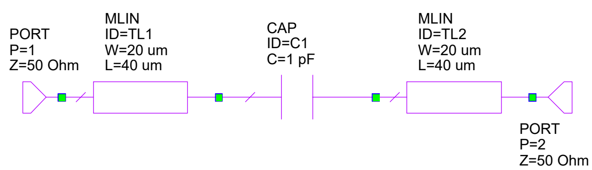
You need to determine what files your manufacturing process requires for your design artwork. The AWR Design Environment software supports GDSII, DXF (not every construct), and Gerber (RS274X format) file formats. You must configure an export mapping table for the output type you want. See “Setting Up Export File Mapping ” for information on how to set up exporting mapping for the file type you want. If you are not familiar with Gerber formats, Gerber file formatting is a good reference.

Drill holes are an area you should study carefully. The AWR Design Environment platform has a drill hole object that you can use to export NC drill files, however it only supports one drill type. These objects are also not draw in Layout View by the models, so you must manually place the drill objects anywhere there is a drill location. Because of these limitations, Cadence suggests using a Visual Basic script written to export drill files from shapes on any layer in your layout. You can find the "Export_PCB_Drill_Gerber" script and associated article with directions for using it by searching for "generates Gerber" in the Cadence AWR Knowledge Base.

B.10. Layout Face Inset Options

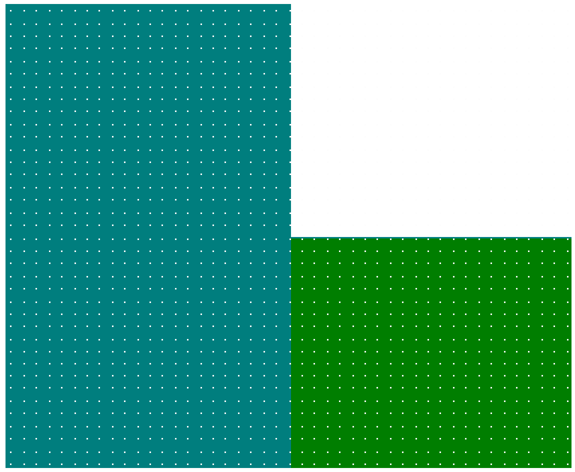
Because the layout system stores shapes on a grid, there is shape point rounding when shapes are rotated at other than 0-, 90-, 180-, or 270-degrees. This rounding can cause gaps in the layout. The gap size is very small, on the order of 1 data base resolution size. See “Determining your Database Resolution” for details on the database resolution setting. Typically, these gaps don't cause manufacturing issues, however, they can. More commonly, the gaps cause problems when performing Boolean operations on your layout when exporting for manufacturing. Cadence commonly sees this problem when exporting while using positive and negative layers; small gaps prevent the shapes from merging properly.

The following simple schematic and its corresponding layout provide an example.

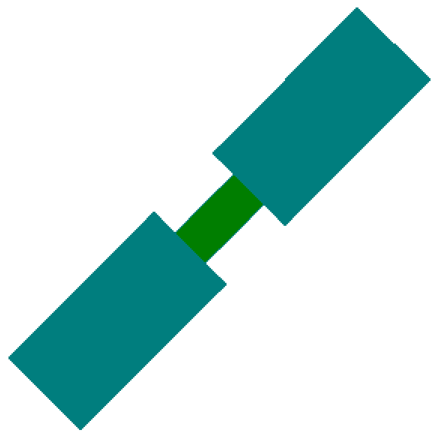


When the layout is not rotated,

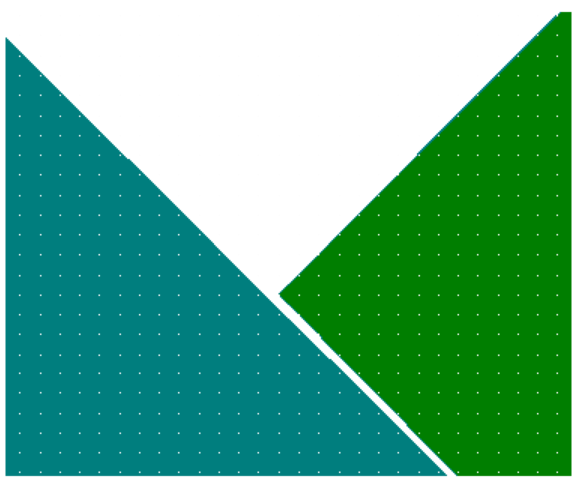
you can zoom in where the models snap together to see there are no gaps in the layout.



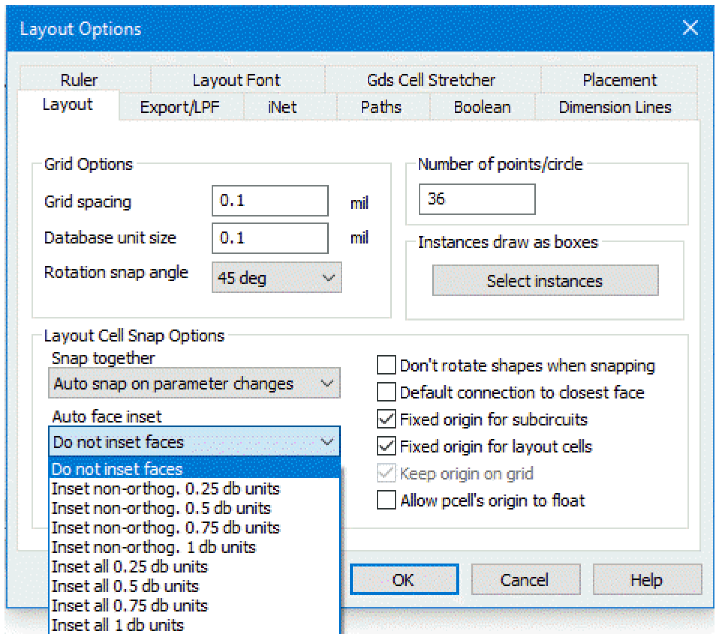
However, if you rotate the layout 45-degrees,



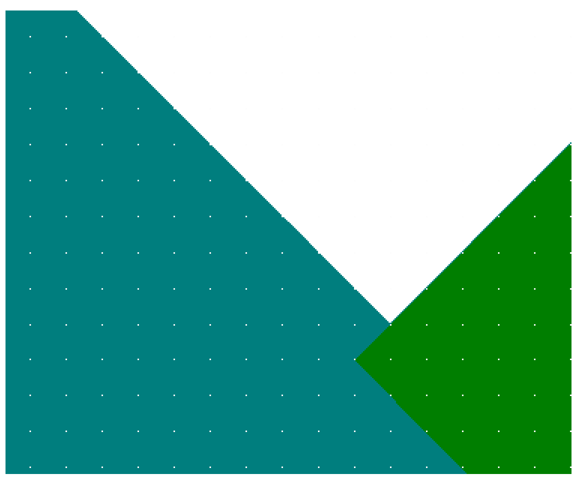
you can zoom in where the models snap together to see there are now gaps in the layout.



You can fix this problem using **Auto Face Inset**. This feature insets the connection faces automatically so there is built-in overlap at the connection points. Cadence recommends choosing a value at the beginning of your design and not changing it. If you change the value later in your design, you must reconnect the entire layout, which may be easy or quite complex depending on the design (this is usually related to the number of closed layout loops in the design). To find the **Auto Face Inset** options, choose **Options > Layout Options** to display the Layout Options dialog box, and click the **Layout** tab. The following dialog box shows the different settings available.



With the faces inset, the gap in layout is gone.

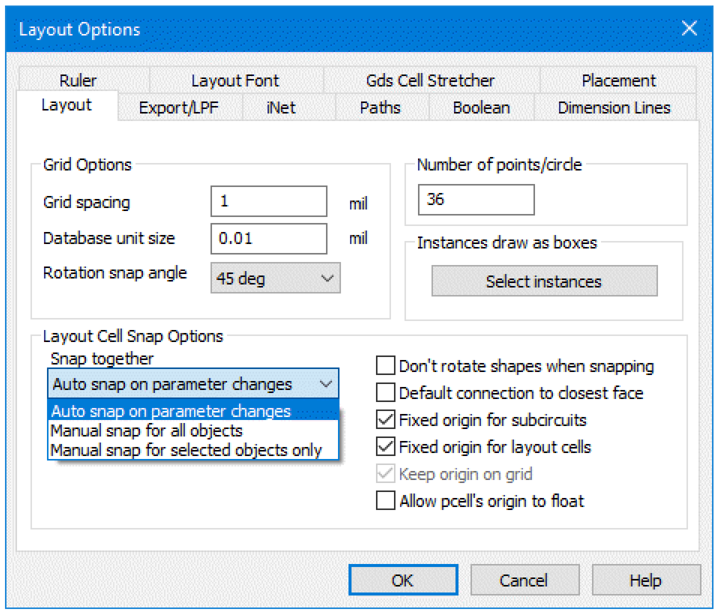


You can experiment with different offset settings. The safest setting is **Inset all 1 db units**. This setting introduces some error into your layout, but the error is at the same order of magnitude as the database resolution for your process. The error is the same value as the error in your processing technology and is typically very small compared to your layout geometries.

You can always check for gaps in your layout using the AWR Design Rule Check (DRC) tool. See "Design Rule Checking (DRC) " for information on configuring and running DRC rules. Depending on the angle of rotation, you can set the inset anywhere from 0.25 db units to 1.0 db units. Another way to potentially create a gap is to use a non-orthogonally rotated version of a schematic in hierarchy. To avoid this you can set **Auto Face Inset** to **Inset all 0.5 db units**. Note that this change affects all faces, not just non-orthogonal faces.

B.10.1. Snapping

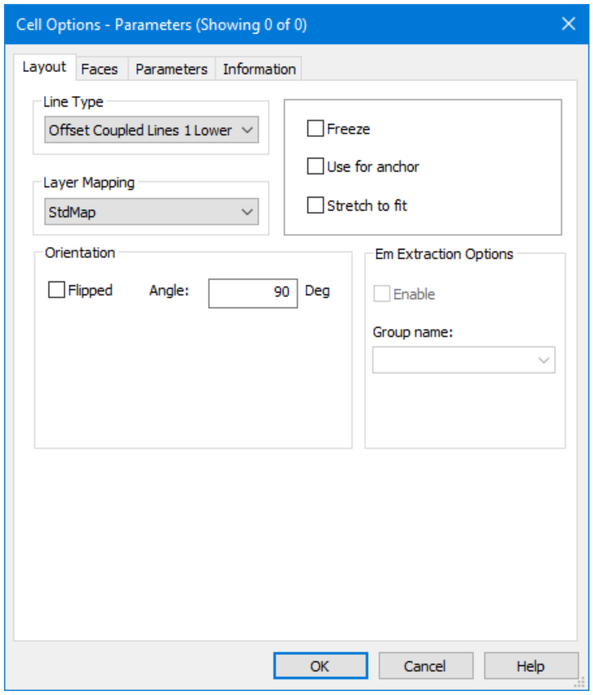
Snapping functions connect layout cell faces in different ways. There are several "Snap together" options to consider when starting a design. To set snapping options, choose **Options > Layout Options** to display the Layout Options dialog box. Click the **Layout** tab to select a **Snap together** option.



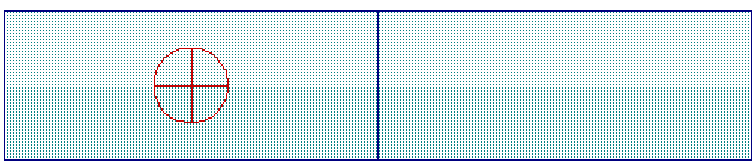
- **Auto snap on parameter changes**: Automatically performs a snap together operation after a parameter that affects layout is changed. This mode is good for simple layouts but can cause problems with more complex circuits, especially those with closed loops. When using this mode, you typically need to use anchors and the freeze function to help control your layout snapping. Some designers are successful with this setting when the design is close to completion, even with closed loops. You can experiment with this setting to see if it works for your design style.
- **Manual snap for all objects**: Snaps together all objects in a layout only when you choose **Edit > Snap Together** or click the **Snap Together** button on the toolbar. This mode is more controlled than auto snap because you must decide when to snap. The same problems arise with complex layouts, however, especially closed loops. When using this mode, you typically need to use anchors and the freeze function to help control your layout snapping.
- **Manual snap for selected objects only**: Snaps together selected objects in a layout only when you choose **Edit > Snap Together** or click the **Snap Together** button on the toolbar. This mode gives you the most control, as not all layout objects are snapped together, only those you select.

You can also snap together your layout hierarchically rather than needing to individually open and snap together each layer of the layout. To use this method, choose **Edit > Snap Objects > Snap All Hierarchy**. This method works only when **Snap Together** is set to **Auto snap on parameter changes** or **Manual snap for all objects**.

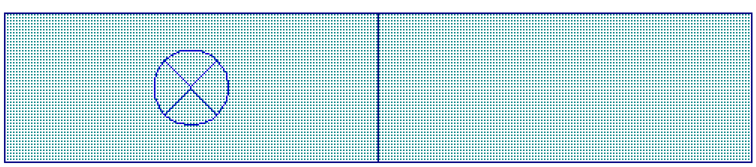
Descriptions of snapping operations also include anchoring and freezing layout objects. To access these settings, right-click a layout object and choose **Shape properties**. The following figure shows an example for one type of layout object; the display for other objects may differ.



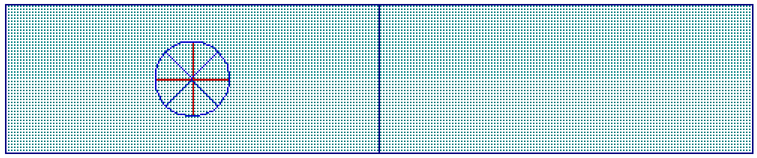
Use for anchor specifies that the snapping commands start with any anchored objects. When an object has an anchor set, a red circle displays in the center of that shape.



Freeze specifies that the snapping commands not move these objects. When an object is frozen, a blue circle displays in the center of that shape.



When both options are applied, the object displays both a blue and red circle in the center of that shape.



B.11. Export Options

It is important to understand layout options if you plan to export your layout. To specify **Layout export options**, choose **Options > Layout Options**, and then click the **Export/LPF** tab on the Layout Options dialog box. You do not need to select these settings at the beginning of a project, however, you should consider them prior to tape-out.

There are potentially four different outcomes for layout export options depending on which combination of check boxes you select:

- If you select neither **Union layout shapes** nor **Subcircuits as instances**, subcircuit layout instances are "flattened" but artwork cells are not, and nothing is unioned. Subcircuits allow you to construct hierarchical circuits by including a circuit block within a schematic.
- If you select **Union layout shapes** but not **Subcircuits as instances**, subcircuit layout instances are "flattened" and unioned, but artwork cells are not.
- If you do not select **Union layout shapes** but you select **Subcircuits as instances**, subcircuits are exported as instances; that is, subcircuits are not "flattened". Artwork cells are treated in the same manner.
- If you select both **Union layout shapes** and **Subcircuits as instances**, subcircuits are exported as instances. Artwork cells are treated in the same manner; however, only elements within a subcircuit are unioned.

There are also four **Artwork instance export options** that address the potential issue of a duplicate cell name:

- If you select **Do not change cell names**, you can only write one of the cells. For instance, if there are multiple cells (from different libraries) that have the same cell name in a layout, only one of the multiple instances is written out to the GDSII file when this option is chosen.
- **Append number to duplicates** allows multiple cell instances with the same name to be written into a single GDSII file. The name of cell instances that were previously written is changed by appending a number preceded by an underscore. For example, if a CellA is written and another CellA needs to be written to the same file, the second instance of CellA is automatically renamed to CellA_1.
- **Append lib name to duplicates** allows multiple cell instances with the same name to be written into a single GDSII file. This option automatically changes the name of cell instances that were previously written out by appending the name of the library from which the cell instance was located. For example, if a CellA from LibA is written, and another CellA from LibB needs to be written to the same file, the first instance is renamed "CellA_LibA" and the second instance is renamed "CellA_LibB".
- **Append lib name to all** appends the library name to all cell instances that are written. For example, CellA from LibA is renamed "CellA_LibA" even if there are no other cell instances named CellA.

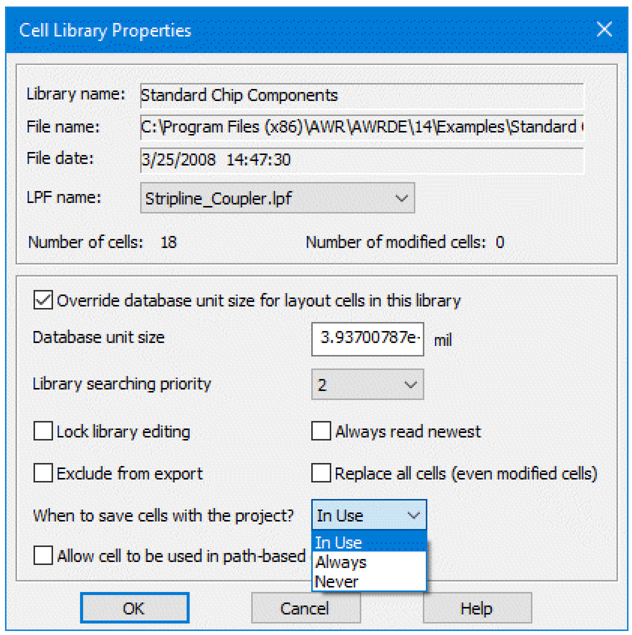
Layout cell export options apply to any object drawn from a parameterized cell (pCell). Pcells are layout representations that use the parameter values of the electrical components to render the layout representation. Microwave Office software has pCells built in for most of the standard microstrip, stripline, and coplanar waveguide components. These options also apply when a layout is exported with hierarchy. **Export flattened** exports pCells flat. **Export all as instances** ensures that all pCells are exported as instances, that is, as separate GDSII or DXF cells for each unique instance of a pCell.

Export specified as instances requires the use of the **Select cells to export as instance** button to select pCells by name for exporting as instances.

Write layout cell parameter map provides the option of writing a text file to the same location as the exported artwork that serves as a map between individual pCells and the instance they use in the exported file. As discussed for the **Layout cell export options**, the **Select cells to export as instance** button only applies if **Layout cell export options** is set to **Export specified as instances**. **Select cells to export as instances** displays a Select Layout Cells dialog box to allow selection of specific pCells to export as instances. Note that your layout must be open to list these pCells.

B.12. Specifying GDSII Cell Library Options

Cadence highly recommends that you familiarize yourself with the various Cell Library Properties dialog box options. To access this dialog box, right-click an existing library name under the **Cell Libraries** node in the Layout Manager and choose **Cell Library Properties**. While each of the options is important, the **When to save cells with the project?** option is discussed in detail.



If you create a library in a project, this option is set to **Always** and cannot be changed, so all of the library cells are saved in the project whether or not they are used. Otherwise, if you import a library into your project, **InUse** is the default setting, although you can change it to **Always** or **Never**.

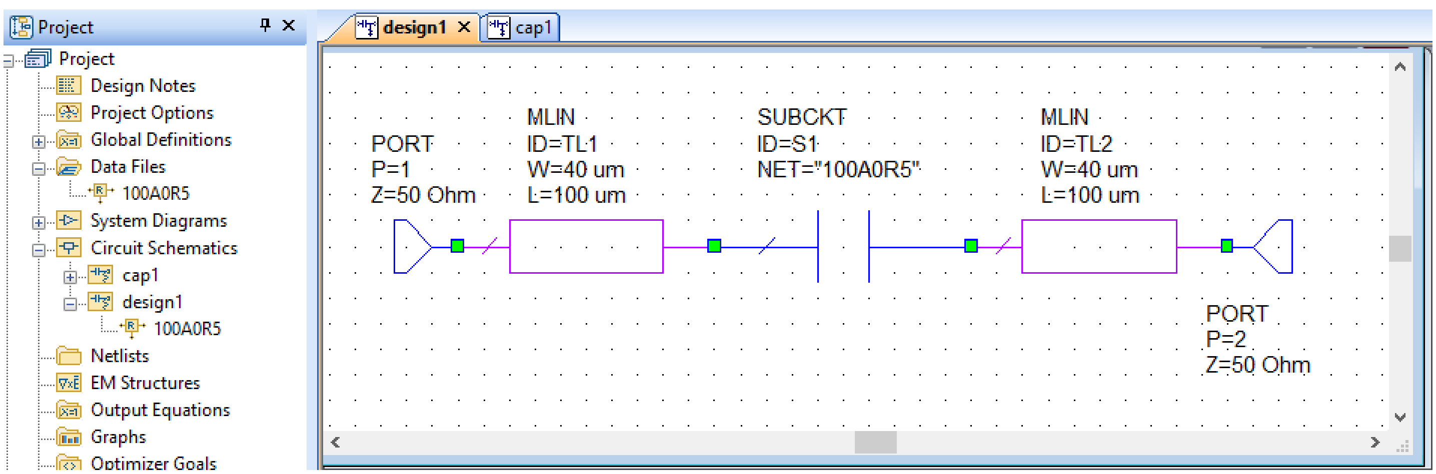
- **InUse**: Only the library cells used in schematic layouts in the project are saved with the project. When the linked file is still available, all the cells from the library are available for use in the project. However, when the linked file is not found, only cells that have already been used in the project are available. This option is useful for keeping project sizes small, as GDSII libraries. The disadvantage is that if you are sending your project to other designers or Cadence Support, the project sent is different than the project on your computer.
- **Always**: All of the cells created in the project are saved in the project. This option is useful for keeping all of the cells from the library available when you send the project to other users. The disadvantage is that GDSII files can quickly make your projects very large.
- **Never**: Cells you create are not saved in the project. This option is useful when using a PDK, and the GDSII cell comes from the PDK. This setting ensures that you always have the latest cells available from the PDK.

B.13. Performing LVS Analysis

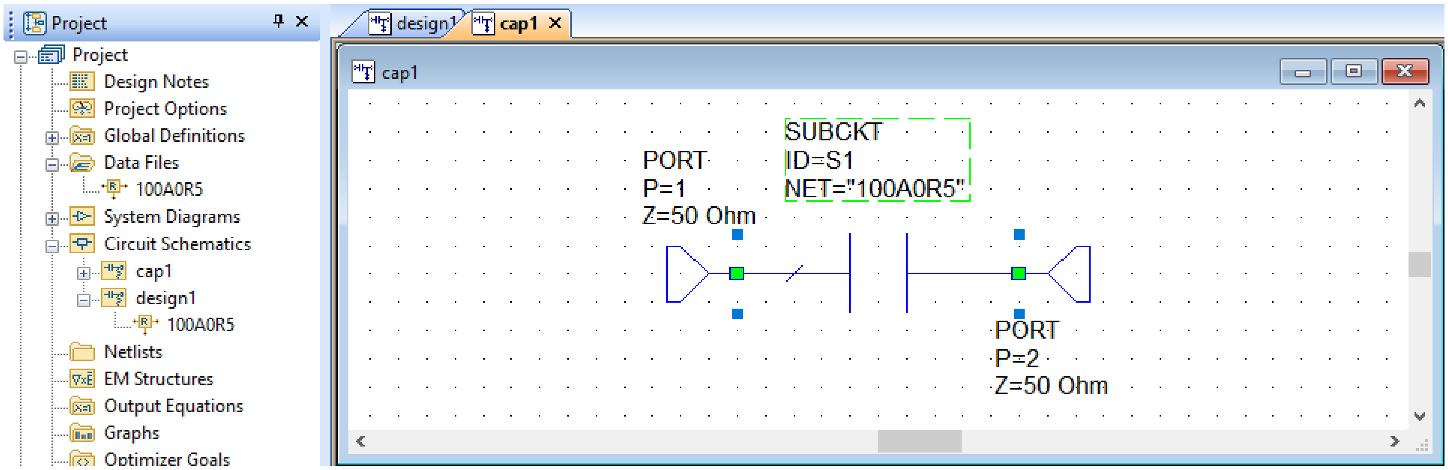
You can perform Layout Versus Schematic (LVS) analysis on projects created in the AWR Design Environment platform. You should be aware, however, that an LVS engine may not understand all portions of your design (such as S-parameter Touchstone data files or EM structures). If you decide to use either of these as subcircuits in your design, you will generate errors when performing LVS.

Historically, one strategy for avoiding these errors was to perform LVS on a "DC schematic" consisting of resistors, capacitors, and devices. However, this may not be a complete representation of your actual design. In addition, this process is time consuming and potentially error prone. A better approach is to use Switch Views for elements that LVS does not recognize. Switch Views are alternative model views or model implementations. For more information about Switch Views and Switch Lists, see “Switch View Concepts ”. Cadence also provides an example named *Switch_Views.emp* that walks you through the process of creating Switch Views and Switch Lists. This example is available in the program */Examples* directory. If you are going to use Switch Views, this example can potentially change how you build your designs.

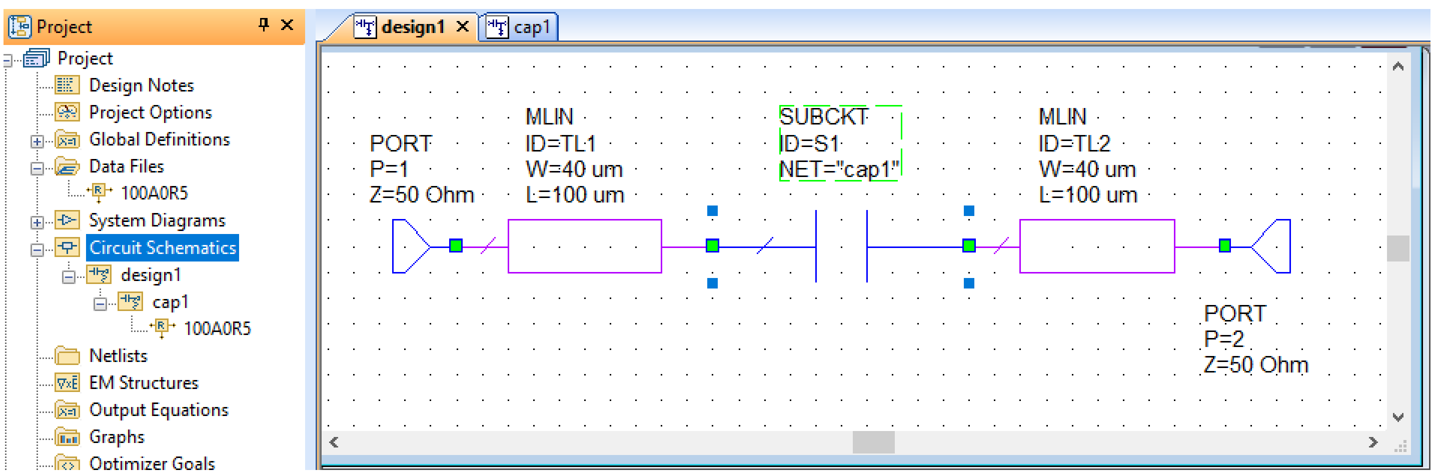
Switch Views can only "switch" entire schematics. If using EM structures, data files, or netlists as a design component, you must add an additional level of hierarchy to be able to apply a Switch View to these components, as shown in the following example. Consider this schematic where the capacitor is an S-parameter data file.



In this configuration, the S-parameter data file is used directly in the design. To properly use Switch Views, you must first make a new schematic and include only the S-parameter data file in the schematic.



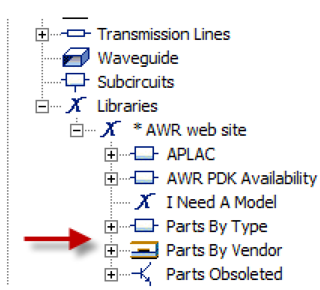
Afterwards, in your design, you use the newly created schematic as the capacitor model.



Now, you can create a Switch View for the "cap1" schematic to properly generate an LVS netlist.

B.14. Component Libraries

Many customers create Printed Circuit Board (PCB) designs that require components to be purchased and mounted on the PCB to manufacture their product. Models and layout cells are required to properly create such a design. You need to determine how you want to create and manage such a library. Cadence provides a library of vendor components in the AWR Design Environment platform as follows:



This data is collected from vendor websites to produce this library. The quality of the library is determined by the information the vendor provides. Cadence highly recommends that you verify each model is correct because the AWR Design Environment platform libraries may not be the latest. You can also contact the vendor for information about their model generation approach. Contact Cadence Support if you find an issue with a library.

Many companies choose to generate their own parts libraries so they can make sure the electrical models are accurate and the layout footprints adhere to their manufacturing process. See Appendix A, *Component Libraries* for details on the AWR component library format and for suggestions on managing these libraries.

Appendix C. AWR Design Environment Errors and Warnings

The Cadence® AWR Design Environment® platform Status Window displays information about various operational and simulation processes in the AWR Design Environment platform. This appendix provides supplemental information and potential solutions for many of the common error or warning messages that display. This information is also accessible directly from the error or warning in the Status Window when a link is provided.

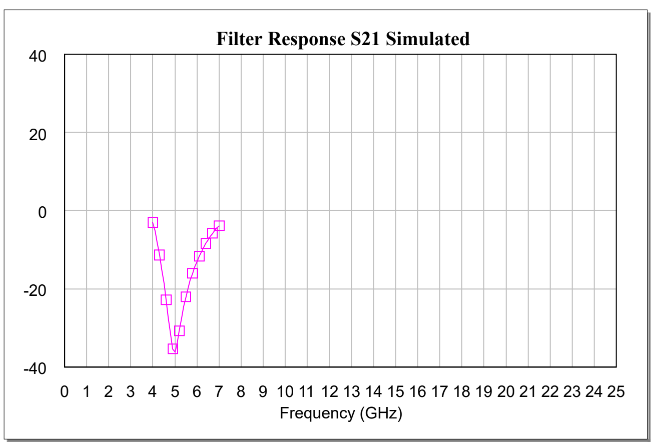
C.1. Extrapolation

Extrapolation warning messages are generated when simulations need network parameter data (either from a data file or an EM simulation) at frequencies outside of the data range provided. Because "extrapolation" means "to intelligently make up data", you should not take these warnings lightly. The following are common problems:

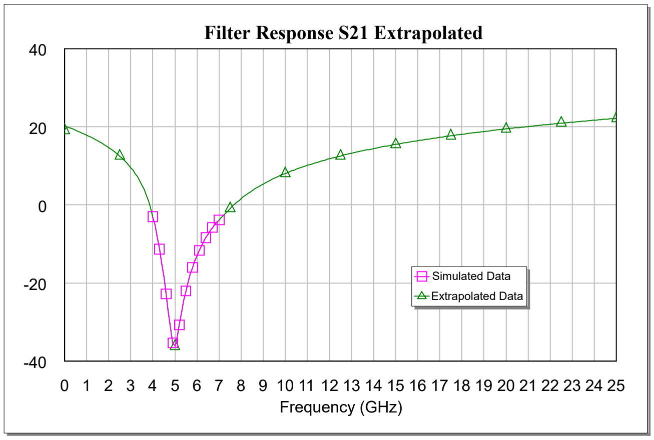
- Bad values at DC which either cause the circuit to bias up wrong or cause convergence problems.
- Bad values at harmonics of the fundamental during harmonic balance (HB) simulation that can output wrong answers or cause convergence problems.

NOTE: During harmonic balance simulations, the highest frequency needed from these files is the highest frequency and the highest harmonic of that frequency set up for simulation. This is why you might see messages regarding extrapolating to frequencies above the highest frequency you specified.

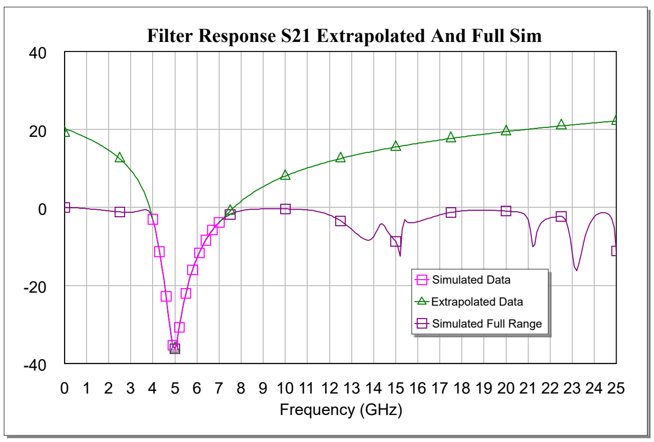
For an example of this problem, see the following figure of an EM simulation (4 to 7 GHz) filter response.



When this EM structure is used in a schematic over a frequency range from 0 to 25 GHz (the frequency range that harmonic balance needs for a 1-tone harmonic balance analysis at 5 GHz with 5 harmonics) the same response is plotted, as shown in the following figure.



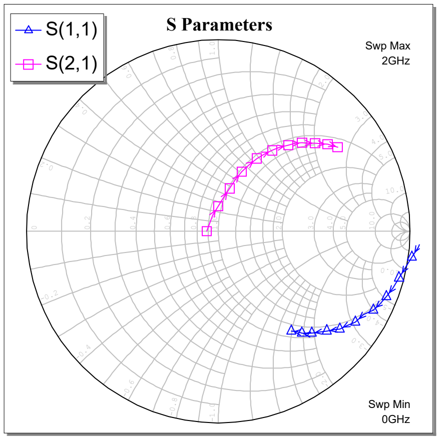
In this very extreme example, notice the response at DC and higher frequency from this PASSIVE circuit. The following response shows the same two curves and the EM structure simulated over the full range.



This example demonstrates that you should not ignore this warning message if it displays.

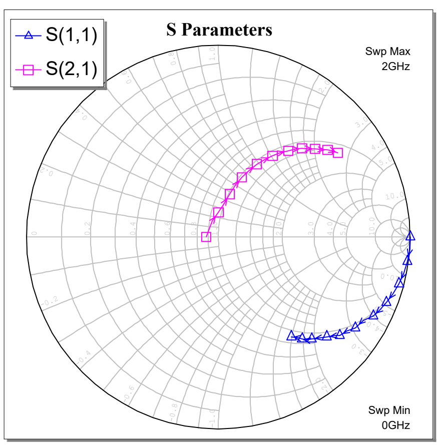
If your subcircuit origin is the Cadence AXIEM® 3D planar EM simulator or EMSight simulator, Cadence recommends that you simulate your EM structure over the full range of your simulation (remember that for harmonic balance the full range is 0 to your highest swept frequency, times your number of harmonics). For specifics on AXIEM analysis at low frequency see “Solvers” and for specifics on EMSight at low frequency see “Low Frequency (DC) Solution ”.

If your data is an S-parameter data file or you do not want to change your EM structure frequencies, the following example shows several possible solutions. In the example, there is a data file for a capacitor with the lowest frequency, 400 MHz. The following plot shows s11 and s21 for this capacitor from 0 to 2 GHz.

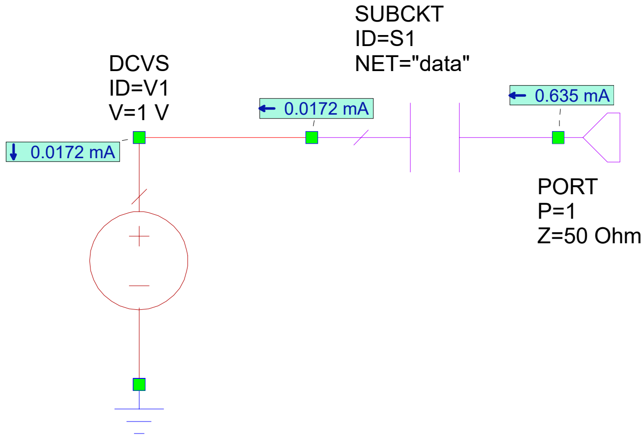


Notice that S11 at DC is outside the Smith Chart.

The first thing to try is to change the interpolation options to see if it fixes the problem. You can change the global interpolation options by choosing **Options > Project Options** and clicking the **Interpolation/Passivity** tab on the Project Options dialog box. Access the Help for this dialog box for information on the different settings. You can also set the interpolation method for individual data files or EM structures by right-clicking the data file or EM structure node in the Project Browser and choosing **Options**. Click the **Interpolation/Passivity** tab on the Options dialog box. The following plot shows the same response when changing from linear to rational function interpolation method.



Notice the data looks better, s11 is not outside the Smith Chart at DC. However a DC current test as shown in the following figure by using a DC source and the DC current annotations still shows that DC is not perfect.



This error may or may not be acceptable to a designer.

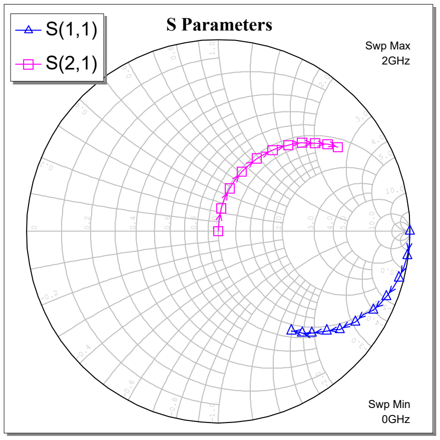
Another possible solution is to correct the data file manually or by writing some code to do so. This requires some knowledge of what the data represents, because, for example, the DC of a capacitor is much different than an inductor. In this example, since we know this is a capacitor, you can add the proper data for the DC point. The original data is shown in the following file for the first few points.

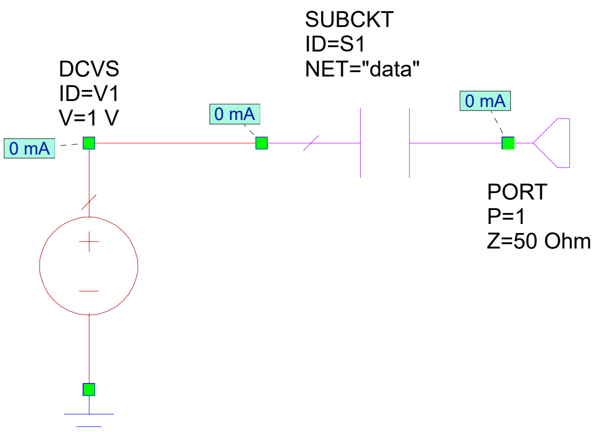
```
# GHZ S DB R 50
! Freq    |S11|    S11 Ph   |S21|    S21 Ph   |S12|    S12 Ph   |S22|    S22 Ph
   0.400    -0.23   -14.46   -12.70    75.15   -12.78    75.23    -0.31   -14.30
   0.499    -0.38   -18.04   -10.93    71.75   -11.00    72.12    -0.45   -16.92
   0.598    -0.53   -21.23    -9.52    68.67    -9.58    68.89    -0.59   -20.14
   0.697    -0.70   -24.41    -8.36    65.40    -8.42    65.42    -0.77   -23.74
```

You can add the following line:

```
# GHZ S DB R 50
! Freq    |S11|    S11 Ph   |S21|    S21 Ph   |S12|    S12 Ph   |S22|    S22 Ph
    0.00     0.00     0.00 -999.00     0.00 -999.00     0.00     0.00     0.00
   0.400    -0.23   -14.46   -12.70    75.15   -12.78    75.23    -0.31   -14.30
   0.499    -0.38   -18.04   -10.93    71.75   -11.00    72.12    -0.45   -16.92
   0.598    -0.53   -21.23    -9.52    68.67    -9.58    68.89    -0.59   -20.14
   0.697    -0.70   -24.41    -8.36    65.40    -8.42    65.42    -0.77   -23.74
```

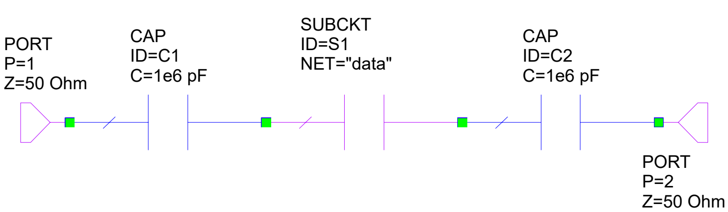
When you add this line the Smith Chart looks good and so do the DC annotations.



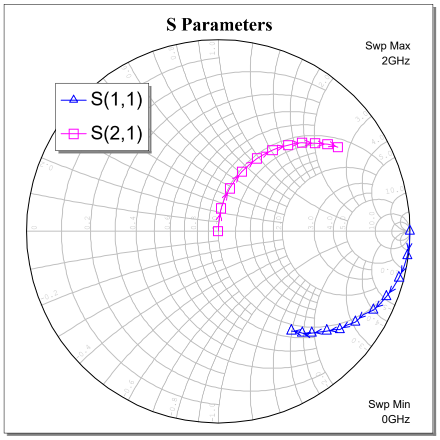


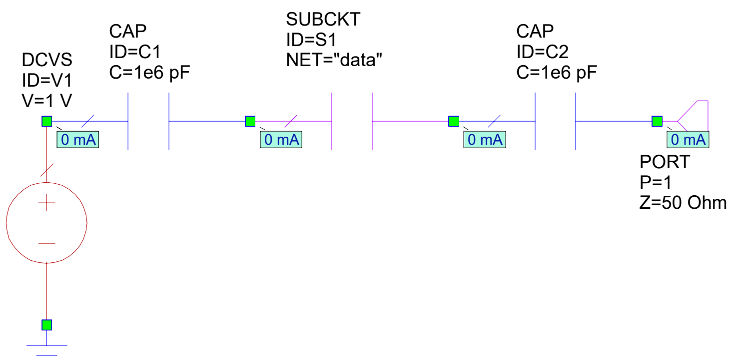
If your data is an EM structure, you can create an S-parameter file for the data using output files. See "Working with Output Files " for more information, and then edit the data.

Another possible solution when the problem is only at DC is to use ideal capacitors and inductors to define the proper DC response of the structure. This again requires knowledge of what the component represents to get the response correct. For this capacitor, we know that DC should be a perfect open, so you can add large capacitors on either side of the data file as show in the following figure.



If this is done in its own schematic, you can use this one subcircuit as the model for that part in the design. For confirmation that this works, see the following Smith Chart and DC annotation for this configuration.





You must be careful with these component values if used in a transient simulation. Very large capacitors or inductors take a long time to reach steady state.

C.2. Cannot Find <item> for the Nonlinear Measurement

Several situations can cause this message, including:

1. A measurement referencing a location in a circuit that no longer exists. Check your circuit again to ensure the measurement location still exists in your schematic.
2. Using the OP_DC or the OP_DYN measurement with the Cadence APLAC® HB simulator. The annotations that are shown are those available from the default AWR harmonic balance simulator and similar values may not yet be implemented for the APLAC simulator.
3. When trying to measure voltage or current on an internal branch of a transistor (for example, for a dynamic load line, IVDLL measurement). The error message that displays is similar to the following:

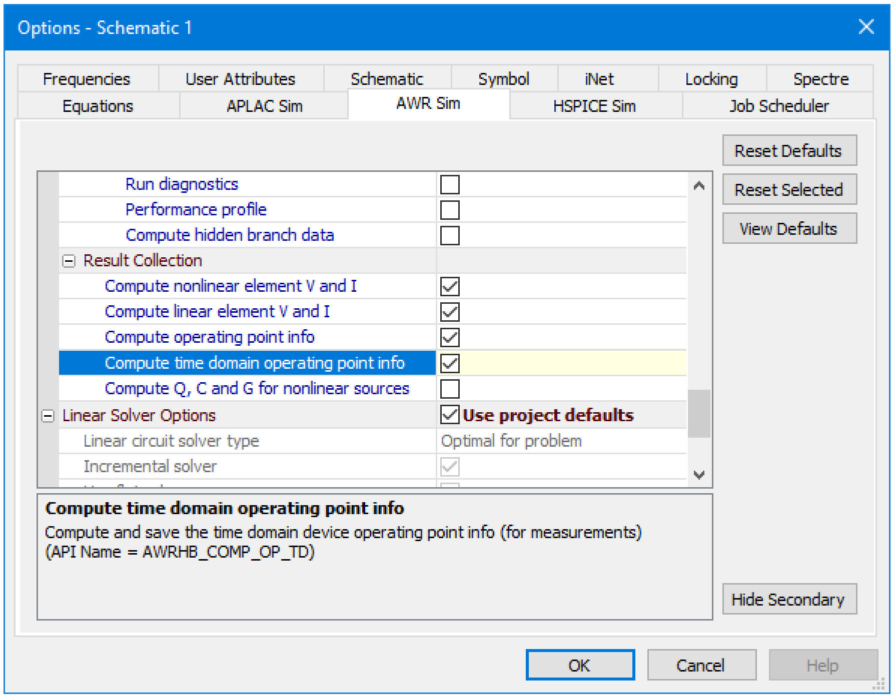
   "[schematic]:[measurement] : Can't find [element]@[branch] for the nonlinear measurement"

   where [item] is the full name of the schematic, measurement, element, or branch. For example, the error message may read:

"Amp:IVDLL(MOSN1A.M1@ds,MOSN1A.M1@ds)[*,*] : Can't find MOSN1A.M1@ds for the nonlinear measurement"

To fix this problem, in the Project Browser right-click the schematic named in the error message and choose **Options** to display the Options dialog box. Click the **AWR Sim** tab. If **Use project defaults** is selected, clear this option.

Under Result Collection, select **Compute time domain operating point info**.



A lot of data is normally created when you analyze large circuits. This may or may not be desirable. By default, nonlinear voltage and current information for intrinsic branches of nonlinear elements is not saved.

C.3. Floating Point Overflow Error in Output Equations

This error is typically caused by a combination of two things:

- The data returned from an Output Equation is a very large number (typically 1.798e308).
- An operation is performed on this number that increases its value (for example, multiplication by some integer).

In the AWR Design Environment platform, Output Equations are identical to measurements on a graph in terms of the data generated by the simulator. The only difference is that in Output Equations, the data is assigned to a variable. There are some situations where data on a graph is not plotted (see the following for details). In Output Equations, this is done by using the number 1.798e308 (the largest double precision number in C++). Any operation that tries to increase the value of this number prompts an error (goes beyond double precision math).

To avoid this error you can use equations to remove these numbers from the data before performing math operations with them.

Output Equations might have this value for:

1. Partial simulation results - when performing a power sweep where the simulator cannot converge at higher power, the points where it does converge are returned as 1.798e308.
2. VSS cascade measurements - these measurements are calculated across each block so the first entry of this vector is not defined and returns 1.798e308.

C.4. Not Translated to SPICE

Not all models in the AWR Design Environment platform can be simulated with transient simulators. This error occurs if you attempt to use one of these models with a transient simulator.

For some PDKs, custom netlisting code had to be written. If you open a second PDK in an AWR Design Environment platform session that has custom netlist information, it is not recognized or used, and this error results. The solution is to shut down the program and reopen it, then load the PDK of interest. If this solution does not work, the model in your PDK may not have been translated yet.

C.5. Step Size for Source Stepping has Decreased Below a Minimum Allowed Value

This error message indicates that harmonic balance did not converge.

To correct this problem, you can try the following solutions:

- The AWR Design Environment platform currently has two harmonic balance simulators. You can try switching to the APLAC harmonic balance simulator by changing the **Simulator** setting for each measurement from **Harmonic Balance** to **APLAC HB**. See "Choosing a Simulator Type" for details. There can be model issues when using the APLAC simulator. If APLAC simulation does not converge, see "APLAC HB Simulator Convergence" for more information.
- The AWR Design Environment platform documentation includes information for working through convergence problems. See "HB Simulator Convergence" for more information.

C.6. Error Evaluating Parameter

This error indicates an incorrect parameter value type. In the AWR Design Environment platform, model parameters that are numbers do not have quotes; string parameters do have quotes. When you see this error, double-click the error and the schematic or system diagram that has the model with the bad parameter opens. Zoom in to that area and highlight the model. The following are some common cases where this error occurs.

C.6.1. Intelligent Cell Syntax

This error occurs when a model using intelligent parameter syntax specifies a parameter that does not exist. For more information on intelligent parameter syntax, see "Using Variables and Equations for Parameter Values" .

To fix this problem you should inspect the parameter values of the element in error, including secondary parameters. If the parameter is using the intelligent syntax make sure that the syntax is correct and that the parameter specified exists.

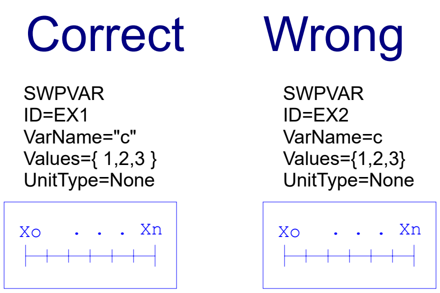
C.6.2. Model Blocks

This error occurs when there is no model block defined for a model that needs the block. The most common case is an MSUB block (that defines the microstrip substrate) for microstrip models. The same error can occur for nonlinear models that point to model blocks. See "Using Elements with Model Blocks" for details on using model blocks for substrates.

Any model that can have multiple matches for a model block generates this error. Previously, the first block found was used, and user operations could change the order. If you see this error when first moving projects to AWR Design Environment software v10 or higher releases, see "Ambiguous Model Blocks" in the AWR Design Environment software v10 "*What's New*" document for details on how to migrate older projects that generate this error.

C.6.3. SWPVAR Blocks

This error can occur with the SWPVAR block VarName parameter. You must enclose the variable name in quotes, as shown in the following figure.

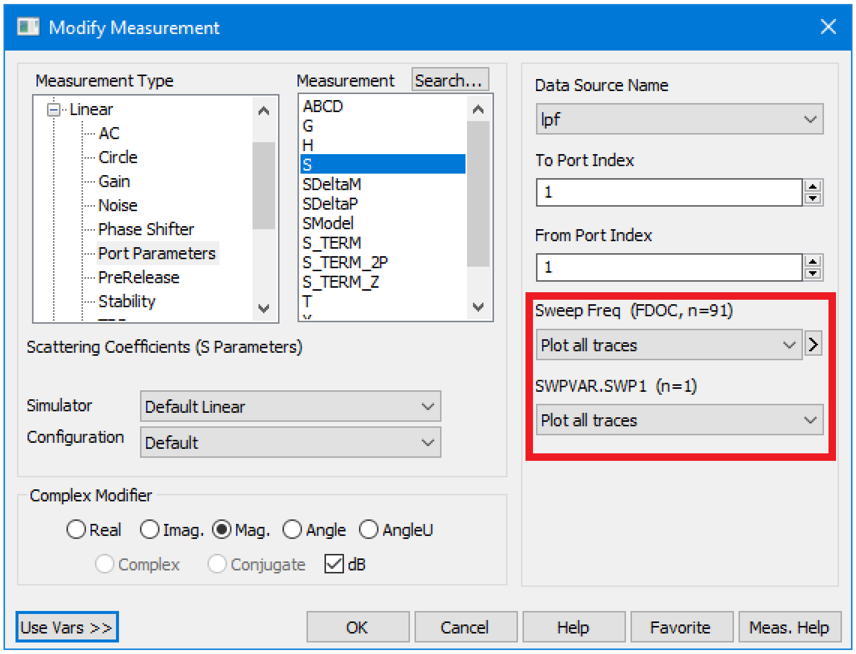


C.7. No Sweep Specified for X-axis

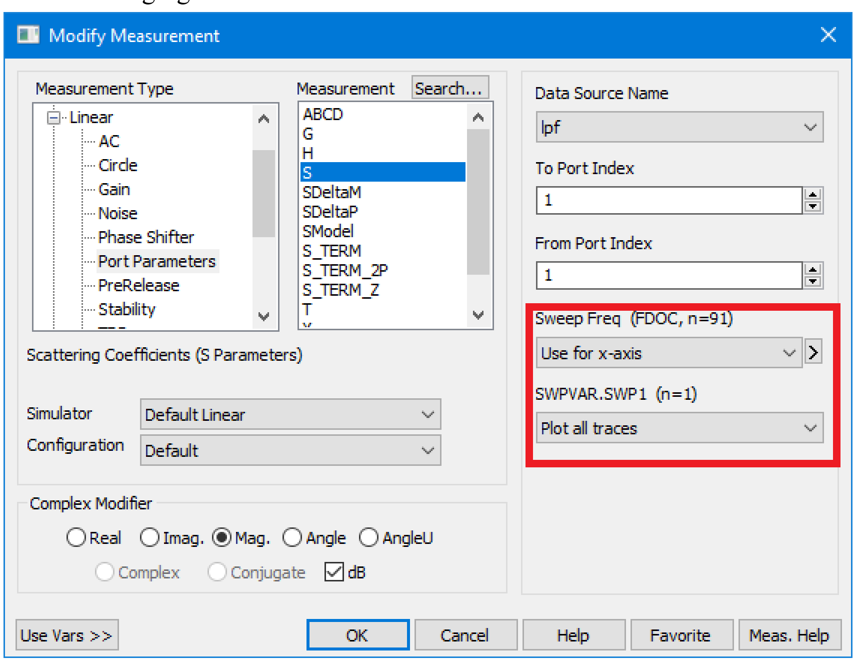
For Cadence Microwave Office® software, every measurement on a graph (regardless of graph type) must specify what is used on the x-axis. Every measurement has frequency as a sweep and then you can add additional sweeps with different elements such as swept sources and the SWPVAR block. For more information on setting up sweeps, see "Swept Parameter Analysis ". These settings might be wrong in the following situations:

- With no sweeps, you try to pick a single frequency from the frequency list.
- You disable or delete a SWPVAR block, and that swept variable is used for the x-axis.

To fix this problem, one of the sweeps must be set to **Use for x-axis**. For example, the following figure shows that neither of the sweeps (highlighted in the red box) uses this setting.



The following figure shows the correction.



If you want to get a single frequency, see “Swept Parameter Analysis ” for options for setting up different frequency sweeps.

C.8. Rise, Fall, and Width Combination Errors

When harmonic balance sources (ports, current, or voltage sources), any rise, fall, and width parameters are set on the source and the frequency is determined by the frequency list defined for the given measurement. For details on how to set up frequency sweeps, see “Swept Parameter Analysis ”. By definition, Time Period = 1/f where *f* is frequency of simulation. If a frequency range is specified, then make sure that TW + TR + TF is less than the period of the highest frequency.

When using a Pulse source, you should set Pulse width (TW), Rise time (TR), and Fall time (TF) such that the sum of these parameters is less than the Time Period of the signal.

When using a Square source, you should set Rise time (TR) and Fall time (TF) such that the sum of these parameters is less than the Time Period of the signal.

When using a Saw source, you should set Fall time (TF) such that the sum of these parameters is less than the Time Period of the signal.

Alternatively, you can use the PORT_SRC, the AC_I, or the AC_V sources that give you more control over how the period and >other parameters of the sources are defined.

C.9. Port Eeff and Gamma Computation Warning for EMSight

For de-embedding length, Cadence generally recommends twice the substrate height or twice the line width, whichever is greater.

The de-embedding algorithm in EMSight uses two structures to determine how to de-embed the embedded simulation. One structure is the length of the drawn de-embedding distance, and a second structure is twice the drawn de-embedding distance. See “De-embedding” for more information. This warning indicates that either the de-embedding length is approximately a quarter wavelength (the second de-embedding structure length is about half a wavelength-full rotation on the Smith Chart), or the substrate is close to multi-moding (it is detecting that the formulas used to get the effective relative dielectric constant and gamma are getting close to singular).

When this warning displays, verify that the measurement results at the frequency specified make sense. If the measurement seems correct, you can disregard the warning. If the measurement looks incorrect, try decreasing the de-embedding length (while changing the enclosure dimensions accordingly). The warning should stop, giving the correct results for the EM simulation.

C.10. Design Rule Violation For X-models

This error message reports: "Design Rule Violation :: Freq should be <= <f> GHz - Strongly Recommend reduced Substrate H or Trace Width to avoid Multimode Propagation."

X-models have an advantage over other closed form models because X-models can issue a warning when being simulated at frequencies where the geometry can support multiple modes. Closed form models are not accurate at frequencies when multiple modes are present; X-models can tell you when you are near such frequencies. For more information on X-models, see “X-models”.

If you need accurate modeling of transmission lines above the frequencies in which the X-models are valid, you need to use an EM simulator such as EMSight or the AXIEM simulator.

The X-models use an equivalent circuit to represent the discontinuity. The values of the components of this equivalent circuit are stored in a table (in the *.emx* file) and interpolation of these parameters is performed over frequency and relative dimensions (w/h). At some frequency point, the transmission line system supports more than one mode of propagation (the higher order modes are not decaying). No node-based circuit simulator is able to correctly simulate a circuit where its transmission lines support more than one mode. Therefore, even if the model could accurately simulate the generalized (more than one propagating mode) S-parameters, the circuit simulator cannot use it. Also, for frequencies where the higher order mode is evanescent, the decay of the mode can be very slow, allowing components to talk to each other through the higher order mode if they are relatively close to one another.

Based on these factors, the X-models only calculate the values of the equivalent circuit up to 80% of the estimated cut-off frequency of the lowest non-TEM mode supported by the largest line entering the discontinuity. For frequencies above this limit, the model issues a design rule warning, and the equivalent circuit parameter for this highest frequency is 0.8*Fc. For frequencies higher than 0.8*Fc stored in the database, an error in the model is shown, although it should degrade gracefully.

Similarly, for dimensions smaller than that supported by the model W/h<0.25, the equivalent circuit parameters for W/h=0.25 are used with first order effect accounted for by dimensional changes. This results in a fairly good model of the reactance, of the discontinuity, for very narrow widths.

Cadence strongly recommends that all transmission lines used in a circuit support only a single propagating mode at the fundamental frequency of the design. Nonlinear components will cause harmonics of the fundamental to be generated. The harmonic balance simulator evaluates the reactance seen at these higher frequencies using the equivalent circuit at 0.8*Fc, as previously mentioned. While this estimate of the impedance from the distributed circuitry prompts an error, this impedance is seen as connected to the nonlinear device itself. Since almost all nonlinear devices have large series inductances and parallel capacitances associated with each port, at high frequencies, the reactance presented by these parasitics dominate the impedance, and thus an error in the estimation of the circuit simulation is tolerable. When attempting to create a circuit (such as a multiplier) all transmission lines should support only one mode for all harmonics used.

C.11. No Frequency Range Defined

This message commonly displays when you import a file as a raw data file type and non-data lines exist at the top of the file. See "Raw Data File Format " for more information about this file format, specifically regarding the comments allowed in these files.

It is not easy to distinguish a raw data file from a Touchstone® file in the AWR Design Environment platform Project Browser. For example, the following text is from the top of a raw data file.

```
# HZ S DB
200000000  -0.546722 0 -99.7851 0 -104.187 0 -0.921447 0
200250000  -0.549285 0 -93.3671 0 -100.488 0 -0.921447 0
200500000  -0.550109 0 -99.1601 0 -94.289 0 -0.922271 0
200750000  -0.555816 0 -107.808 0 -99.7578 0 -0.923461 0
```

The # HZ S DB line is causing this error. This is confusing because a Touchstone file must have this control line. The solution in this case is to remove the non-data lines from the file, so the file displays as follows.

```
200000000  -0.546722 0 -99.7851 0 -104.187 0 -0.921447 0
200250000  -0.549285 0 -93.3671 0 -100.488 0 -0.921447 0
200500000  -0.550109 0 -99.1601 0 -94.289 0 -0.922271 0
200750000  -0.555816 0 -107.808 0 -99.7578 0 -0.923461 0
```

NOTE: The easiest way to determine if a file is a Touchstone file or a raw data file is to right-click the file in the Project Browser and choose **Export Data File**. The default **Save as type** displays the file type, "S-Data Files" for Touchstone or "Raw Data File Port Parameters" for raw data type.

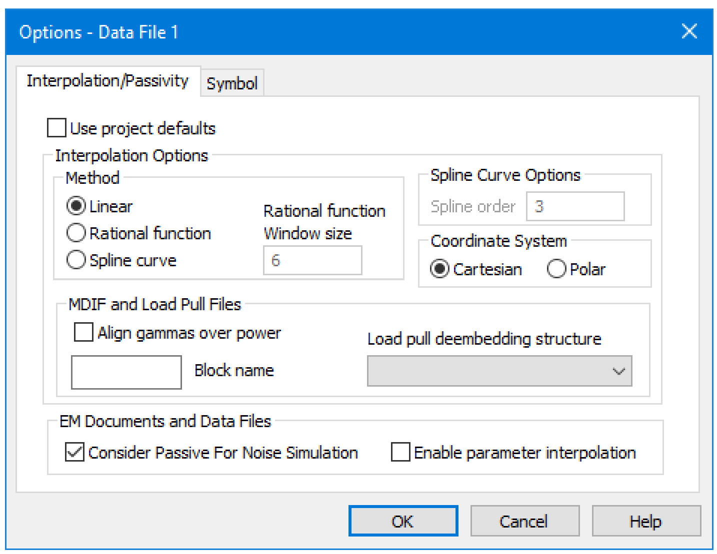
This error may also display if there is a duplicate frequency in the data. You must fix this duplicate frequency. One method is to make a copy of the data and start deleting half of the data until you can plot it, then keep trimming down the data until you can find the problem area.

This error also occurs if your data does not complete a full matrix. The raw data expects the proper number of entries for a 2x2, 3x3, 4x4... matrix of data. For 2x2 data you must have 9 entries (1 for frequency, 4 matrix entries x 2 since data is complex). For 3x3 data you must have 19 entries, and so on.

C.12. Not Passive and Does Not Contain Any Noise Data

Two-port Touchstone data files have two different mechanisms that calculate noise, based on whether the data is passive or not. If the data is not passive, a special section of the file determines how to simulate noise. See "Touchstone File Format " for more information. If the data is passive, the noise is determined directly from the network response. You can view the passivity of a data file by using the PASSIVE measurement, which determines if a file is active or passive. See " Passive: PASSIVE" for details on this measurement.

Sometimes a passive data file can be slightly active due to numerical issues or slight calibration problems. In these instances you can force a data file to be considered passive for noise by right-clicking the data file in the Project Browser and choosing **Options**. In the Options dialog box, clear the **Use project defaults** check box and select the **Consider Passive For Noise Simulation** check box as shown in the following figure.



C.13. Problem with File Format

This error message reports: "Error reading line <line>: Invalid noise data found. Expecting 4 entries after frequency, found 8."

Two-port Touchstone data files can have separate noise parameters for active circuits. While reading these files, look for frequencies that are not in order to locate the beginning of the noise section of the file. This message is usually generated when there are duplicate frequency entries in the non-noise section of the data. The message tells you on which line a problem is detected. You can open the data file and locate this line to check for duplicate frequencies in your file at that location.

C.14. X-model Autofill Message (Understanding X-models)

When using X-models, you should understand these issues:

- Which models are X-models
- How to properly set up substrate parameters for X-models
- How to fill a new database

See "X-models" for more information on these topics.

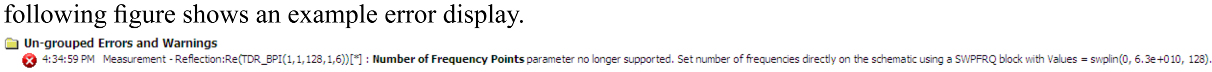
C.15. Time Domain Reflectometry (TDR) Measurement Update

Prior to v9.0, the TDR was calculated from the end points of the frequencies selected and the **Number of Frequency Points** set for the measurement. In v9.0, the TDR measurements use only the frequency list specified for the measurement; the **Number of Frequency Points** is no longer used. Because of this difference, older projects require some changes to make them function as they did prior to v9.0.

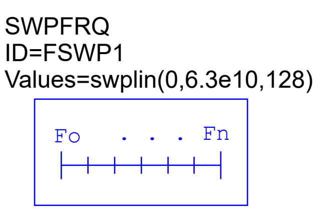
When opening previous projects, you can do one of the following:

- Change the **Number of Frequency Points** parameter on the measurement to 0 so the measurement uses the frequency list specified for the **Sweep Freq** measurement setting. Note that if using a Low-Pass TDR measurement, you must have 0 in the frequency list you use. Also, remember that the frequency step value you use determines the maximum time and the maximum frequency along with the **Time Resolution Factor** parameter that determines the time resolution for the measurement. You can edit these settings until you get close to the result you got prior to v9.0.
- Another option provides the same results as prior to v9.0 but includes more steps:

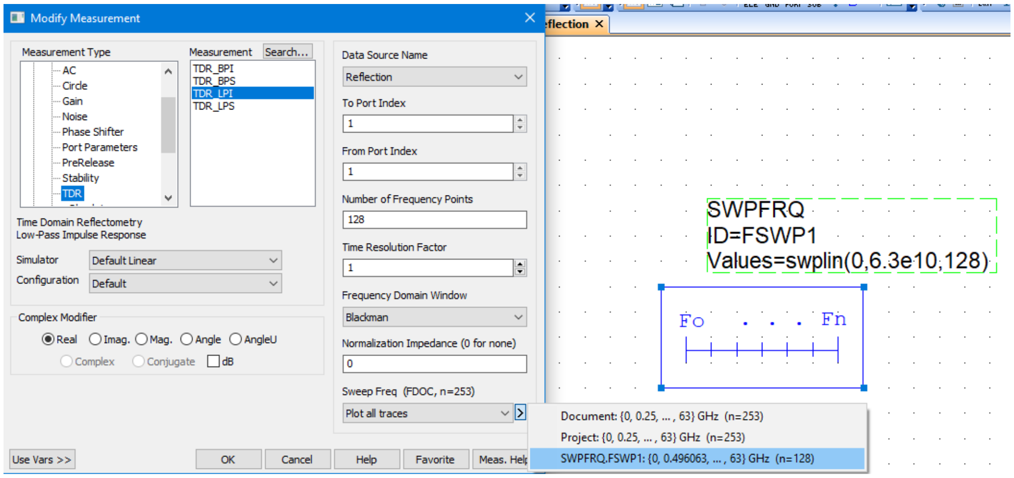
  1. Add a SWPFRQ block to each schematic where a TDR measurement is calculated.
  2. For the Values parameter, use the syntax included in the error listed for each TDR measurement. For example, the following figure shows an example error display.

     Notice that the error lists the exact syntax needed. Based on this message, the SWPFRQ block should appear as follows.

  3. Edit the TDR measurement to use the **Sweep Freq** setting to point to the SWPFRQ block added to the schematic.



4. Change the **Number of Frequency Points** to 0. If this parameter is not 0, an error displays if the number of points set do not match the number of points in the sweep frequency list.

If you perform TDR measurements directly on a data file and do not have a 0 in the frequency list you are using, an error displays, just as with a schematic. To correct this error you can do one of the following:

- Create a new schematic, add a SUBCKT element, and reference the data file using the SUBCKT element. You can configure the frequency range to apply to this schematic only. To do so, right-click the new schematic and choose **Options** to display the Options dialog box. Click the **Frequencies** tab and clear the **Use project defaults** check box. You can now add a 0 to the frequency list in the **Modify Range** section. If you use this schematic (which includes the 0 frequency point) as a subcircuit in place of the actual data file (which does not have the 0 frequency point), no error displays when you select document frequencies **(FDOC)** to use for the TDR measurement. In fact, a warning displays related to extrapolating from the lowest frequency contained in the data file down to the 0 frequency point you added.
- You can also add a 0 frequency point in the Project Options dialog box (choose **Options > Project Options** and click the **Frequencies** tab), although adding the point in this dialog box applies it to all schematics for which **Use project defaults** is selected, as described in the previous solution. In the TDR measurement, you then need to specify that project frequency **(FPRJ)** is used.

C.16. MWOfficePS.dll is Too Old or Cannot be Found

This warning alerts you that your installation might not be correct. As a precaution, Cadence recommends that you repair the installation of the newest version of the AWR Design Environment software that you are running. You can repair the installation by running the installer **.msi* file and clicking the **Repair** option. With administrative rights on your computer, you can also open the Windows Control Panel **Add or Remove Programs** feature, find your AWR Design Environment software installation, highlight it, and then click the **Change** button. An option to repair your installation displays.

C.17. Repairing the AWR Design Environment Software Installation

There are situations where you might need to repair your AWR Design Environment software installation. An installation repair restores the AWR Design Environment software to its originally installed state. This may be necessary if you have two major versions of the software installed (for example, AWR 2008 and AWR 2009) and you uninstall one version. You should repair the other version to ensure the software runs properly.

You can repair your installation using one of the following methods:

Method 1 - Using the **.msi* installer file you used to install the AWR Design Environment software.

To use this method:

1. Double-click the **.msi* file.
2. Choose the option to **Repair** your installation, then click the **Finish** button.

Method 2 - Using the **Add or Remove Programs** utility (you can also use the original installation file for repair purposes).

To do this:

1. Open the Windows Control Panel and double-click **Add or Remove Programs**.
2. Click on the AWR Design Environment software installation you want to repair.
3. Click the **Click here for support information** link to open the Support Info dialog box.
4. Click the **Repair** button.

C.18. Failure Initializing the AWR Scripting IDE Addin

The Scripting Editor can prompt errors for different reasons. Sometimes the older versions of the AWR Design Environment software interfere with the Scripting Editor in more recent versions.

The most common solution is to repair your AWR Design Environment software installation. See "Repairing the AWR Design Environment Software Installation " for the steps to repair your installation.

If an installation repair does not solve the problem, open your program installation directory and double-click the *AddinManager.exe* file to open the Addin Manager. Clear the check box for the **AWR Scripting IDE** and start an AWR Design Environment platform session. Restart the Addin Manager and select the check box for the **AWR Scripting IDE**, then start another AWR Design Environment session.

C.19. Unregistered OLE DLLs

For an unknown reason, the registrations of two of the operating system-supplied DLLs can be corrupted.

To solve this problem, you need to re-register the DLLs, or repair the installation.

To re-register the system DLLs:

1. Open a command prompt (in Windows, choose **Start > Run** and type "**CMD**" in the window).
2. In the command window that opens, type the following lines one at a time and then press **Enter**.

   ```
   Regsvr32.exe OleAut32.dll
   ```

   ```
   Regsvr32.exe Ole32.dll
   ```
3. Restart the AWR Design Environment software.

See "Repairing the AWR Design Environment Software Installation " for steps to repair your installation.

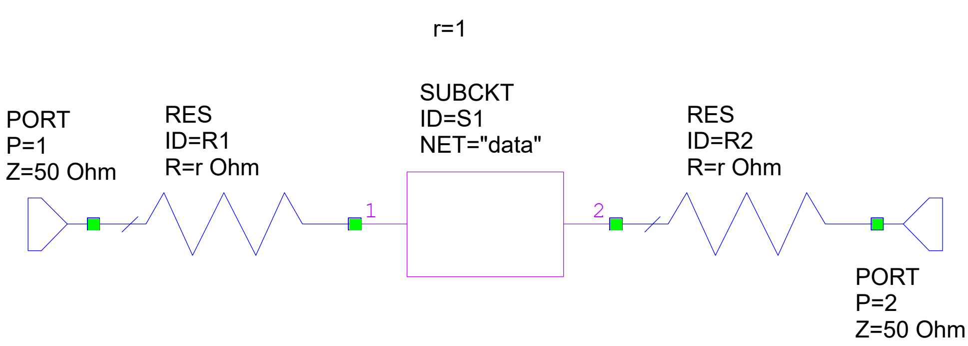
C.20. Active NPort Found When Computing NDF

NDF simulation, as it was originally formulated, modifies (zeros) the Gm of an active device to compute a determinant that is used to determine the stability of the circuit. See "Normalized Determinant Function: NDF " for details. Because there is no way to isolate the contribution of the Gm from network parameter data, the entire N-port is treated as either active or passive for NDF. For an active N-port, the equivalent of zeroing the Gm is to zero the whole N-port. For a passive N-port, however, the treatment should be correct (nothing is zeroed).

Each network data file is checked for its passivity. You can perform the same check using the PASSIVE measurement. See “ Passive: PASSIVE” for details.

To correct this problem, you can try the following solutions:

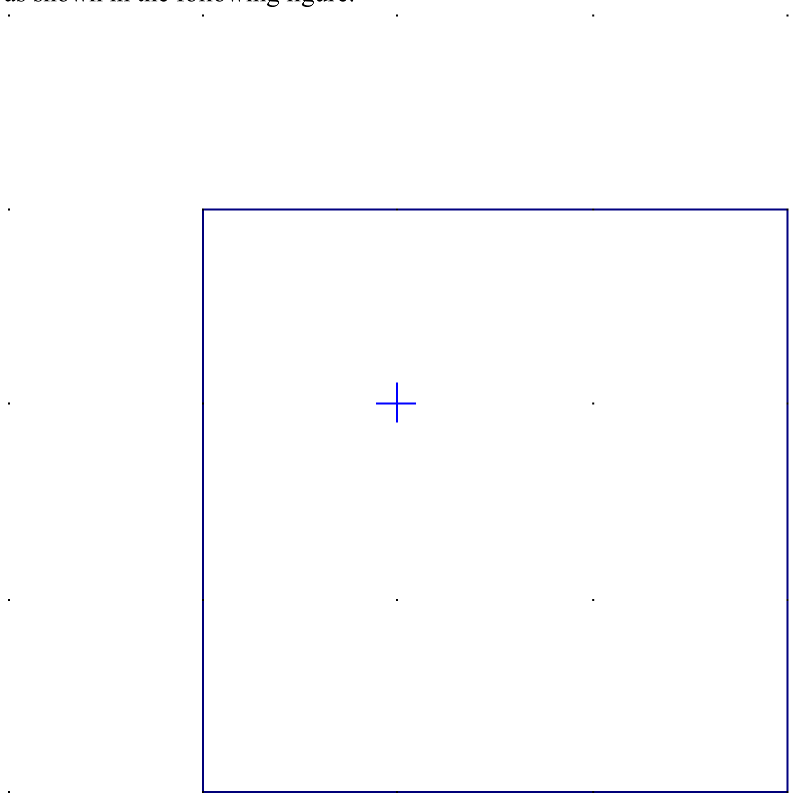
- EMSight: Passivity problems arise from not spacing shapes far enough to the sidewalls or the top or bottom of the enclosure. Try adjusting these dimensions to see if it helps the passivity of the structure.
- AXIEM: AXIEM analysis software includes a built-in passivity correction setting. See “Passivity Enforcement” for details.
- Other EM tools: Work with the EM vendor to determine how to make your EM simulation data passive.
- Other Data: This data is supplied by vendors or measurements taken in the lab where you cannot easily regenerate the data. In this case, Cadence suggests you do the following:
  1. Create a schematic and add the data file as a subcircuit.
  2. Add a resistor between each node of the data file and a port.
  3. Assign the same variable to each resistor and add an equation for the variable used.
  4. Use the PASSIVE measurement on the schematic and then adjust the value of resistance until the data is passive. This gives you an idea of how much loss is needed to make your data passive. You need to judge whether the amount of loss added is acceptable or not. If not, you probably should not use this data in your design.
  5. If the loss added is acceptable, you can use this schematic as your model (with the loss included) or you can create a new data file for this schematic using output files. See “Working with Output Files ” for details. The AWR Design Environment platform provides a utility to help create a new data file. To access this utility, choose **Scripts > Data > Write_Output_File**.



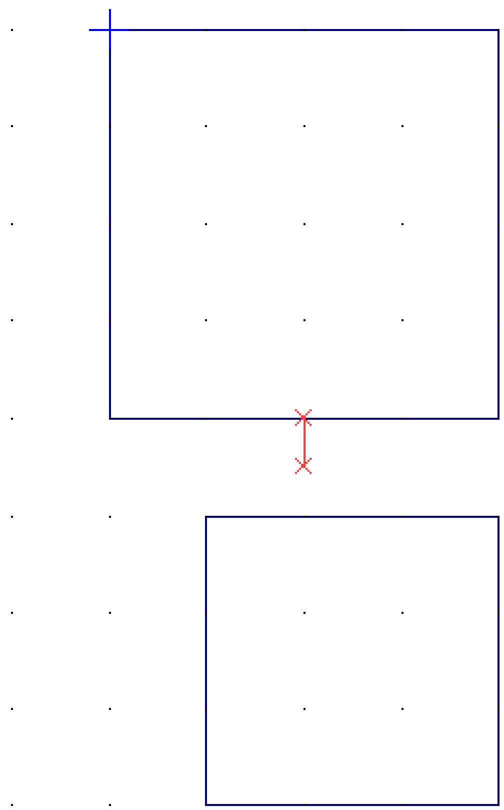
C.21. Area Pins Must be 2x the DBU

This error indicates that there are area pins whose size is not 2x the Data Base Units (DBU) of the design. These area pins can cause gaps in layout when used with hierarchy.

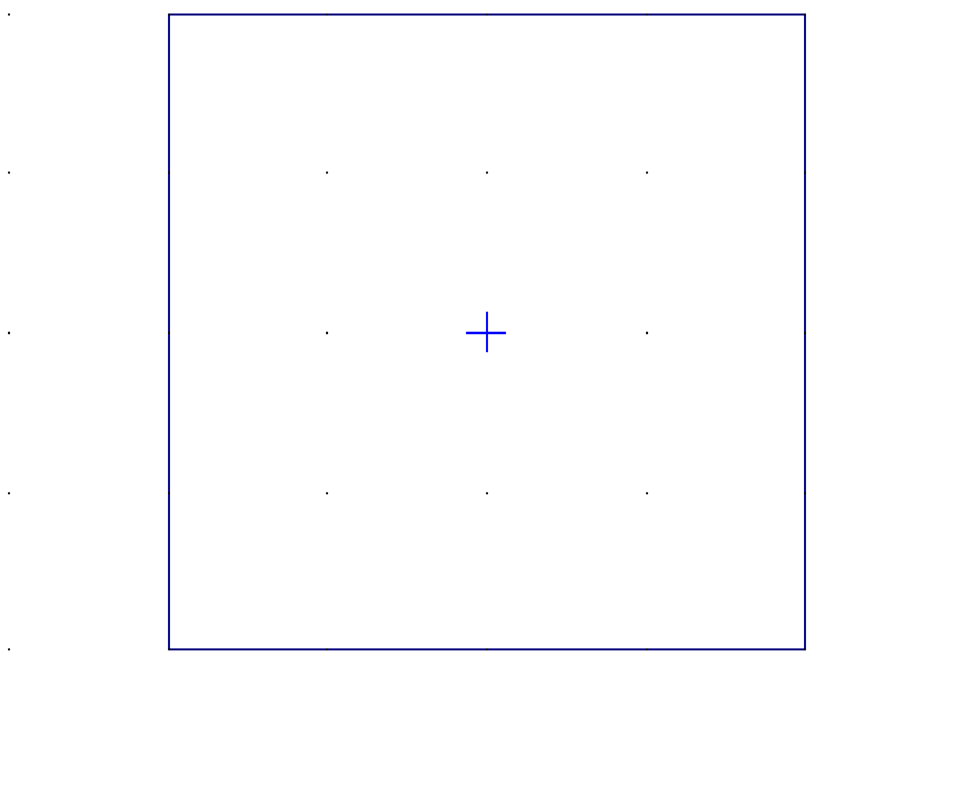
The Data Base Unit (DBU) of a project is the smallest size at which shapes can be drawn. See “Determining your Database Resolution” for details on setting the DBU. The original area pins in the AWR Design Environment platform defined the center of the pin as the origin of the layout cell. Because of the center issue, if the cell is not 2x the DBU, everything gets rounded to put the shape on the DBU. For example, a DBU and grid size of 1um results in a 3um x 3um area pin, as shown in the following figure.



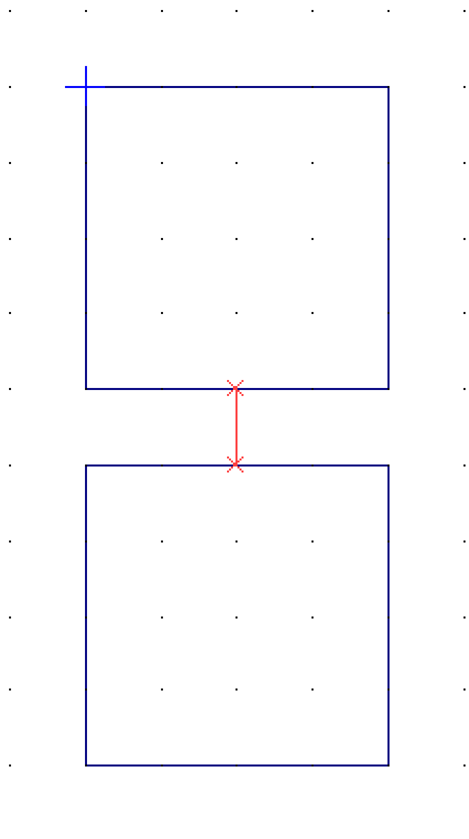
The fill color is turned off for the shape. The cross in the figure is the origin of the area pin. Note that it is rounded so that it is on grid, yet it is not in the center of the square. The following figure shows this area pin used at a higher level of hierarchy and rotated, with an MLIN is connected to it.



Notice that the rat line does not properly connect to the area pin's edge due to rounding. The following figure shows the area pin as 4um x 4um, and the layout of just the area pin.



Notice the cross in the center of the cell. If this area pin is used at a higher level of hierarchy and rotated, and an MLIN is connected to it, it displays as shown in the following figure.



The rat line is now properly showing connections to the edge of each shape.

This problem is commonly found when using the "RECT_PIN" layout cell for ports to define connectivity locations through hierarchy. To fix this for the RECT_PIN, you can:

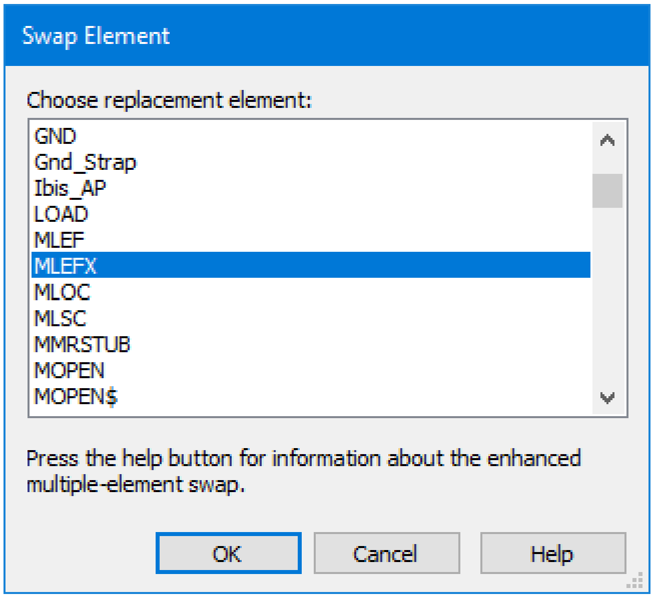
- Resize each of the area pins so they are 2x the DBU. You can find each by double-clicking the error message to open the layout and zoom to the erroneous area pin.
- You can use the RECT_PIN2 layout cell instead. This cell defines its origin on the corner of the area (not the middle), so its size can be 1x the DBU.

C.22. Using MOPENX Model with Secondary L Parameter Not Set to 0

The MOPENX model is equivalent to the MLEFX model (see “Microstrip Open End Effect (EM Base): MOPENX”) with the L parameter set to 0. MOPENX has a secondary parameter for L that you can change, and which changes simulation results. The layout for the model, however, will not be correct. In this case it is best to use the MLEFX model since the layout is correct and the simulation results are identical.

When you get this error message, you can make this change by:

1. Double-clicking the error message to open the schematic containing the model; the model is easily identified in the schematic.
2. Right-clicking the model and choosing **Swap Element**.
3. In the Swap Element dialog box, select **MLEFX**

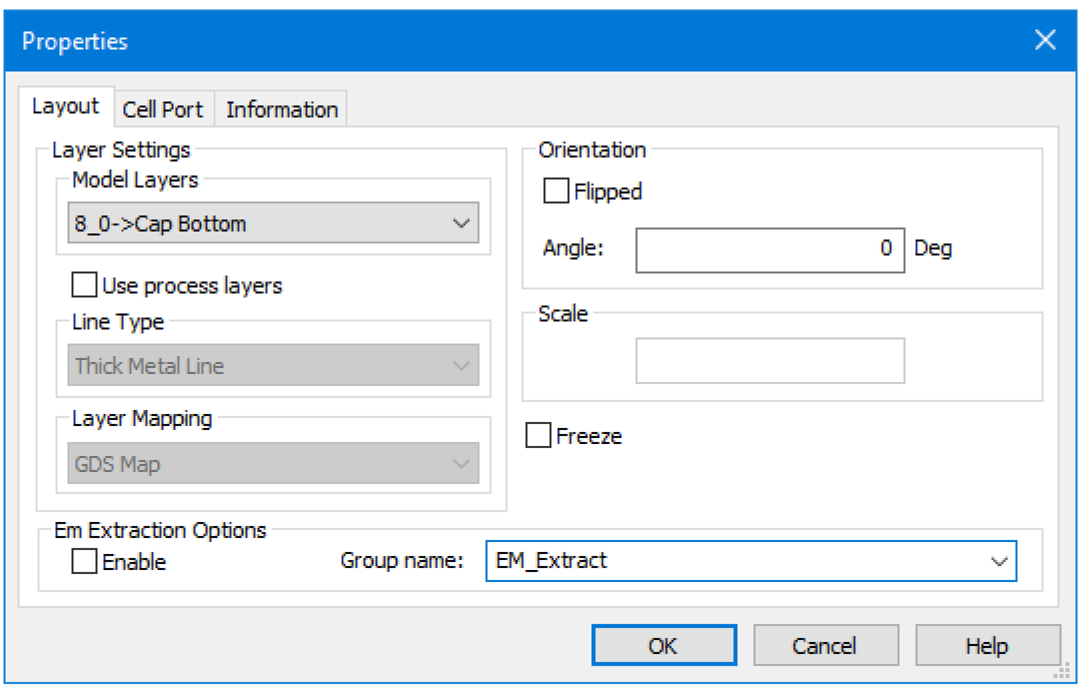


These steps maintain the original settings for the W and L parameters of the original model.

C.23. Port_Number: Face(s) Not on a Drawing Layer

This error indicates that the cell port in an artwork cell is on an unknown drawing layer. Each cell port must be on a drawing layer specified in the LPF.

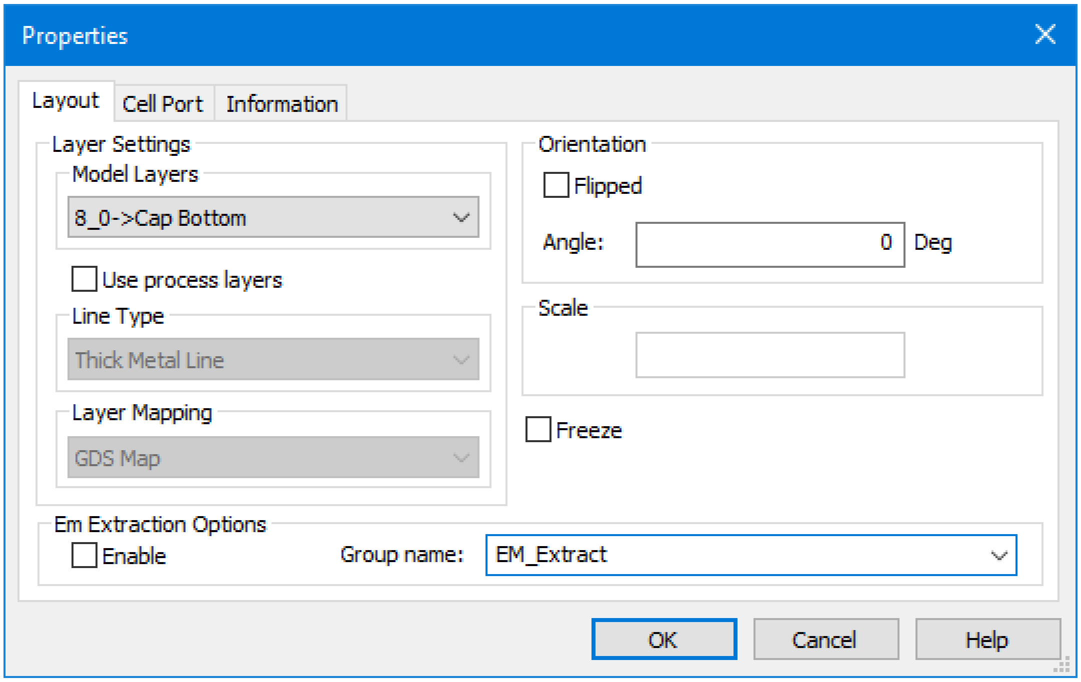
To correct this problem, select the cell port in artwork cell, right-click and choose **Shape Properties** to display the Properties dialog box. On the **Layout** tab, select an appropriate model layer from **Model Layers**. See "Connectivity Checking " for more information.



C.24. Port_Number: Detached Face(s) on Drawing Layer Without Connectivity Rules

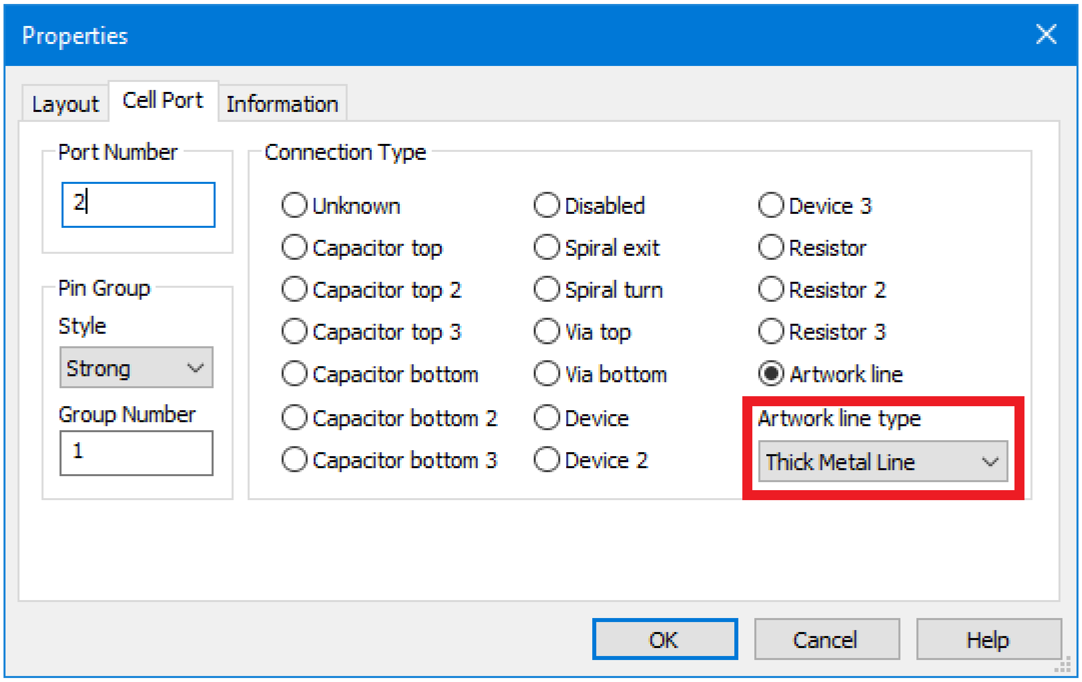
This error indicates that the cell port is not touching any metal inside the artwork cell and is using a drawing layer for which there is no connectivity rule. See "Connectivity Checking " for more information.

Check the artwork cell to make sure the face is at a correct location and uses a drawing layer for which a connectivity rule exists. To check which drawing layer the port is using, select the cell port, right-click and choose **Shape Properties** to display the Properties dialog box. On the **Layout** tab, check the model layer specified in **Model Layers**.

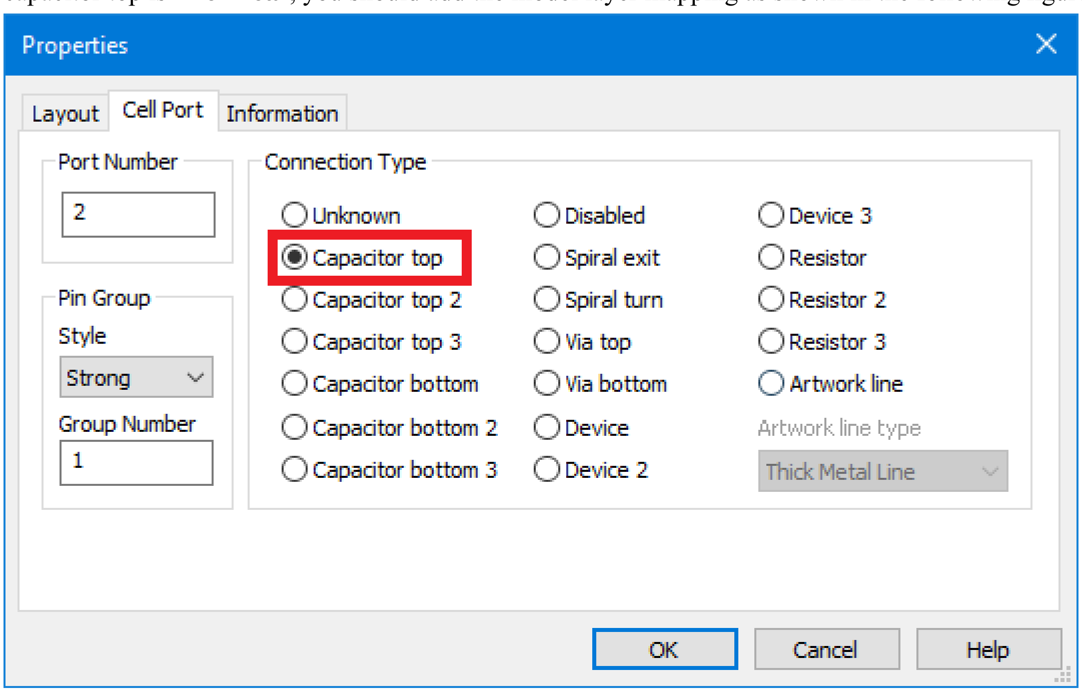


The resolution of this issue depends on the cell port connection type. To determine the cell port **Connection Type**, select the cell port, right-click and choose **Shape Properties** to display the Properties dialog box. Click the **Cell Port** tab and refer to the following to resolve the issue:

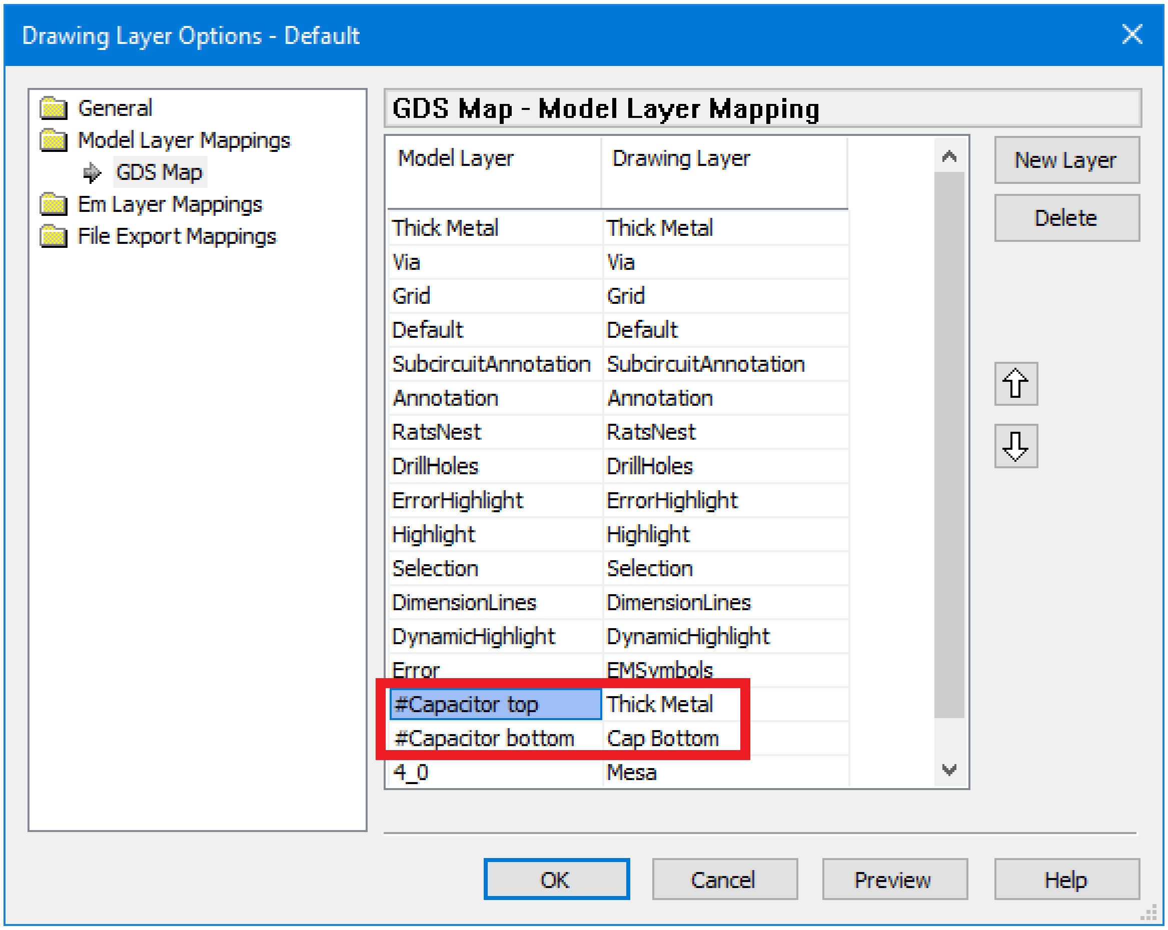
- If **Connection Type** is **Unknown**, the cell port must be on the same layer as the shape to which it is attached.
- If the **Connection Type** is known, but set to **Unknown**, change it to the proper connection type as shown in the following step, or if it is an **Artwork line**, use the correct **Artwork line type** from the drop-down list.



- If the **Connection Type** is other than **Unknown** and **Artwork line** is used, you need to add the model layer mapping for the connection type used. For example, if a port connection type is set to **Capacitor top** and the drawing layer used for capacitor top is **Thick Metal**, you should add the model layer mapping as shown in the following figure.



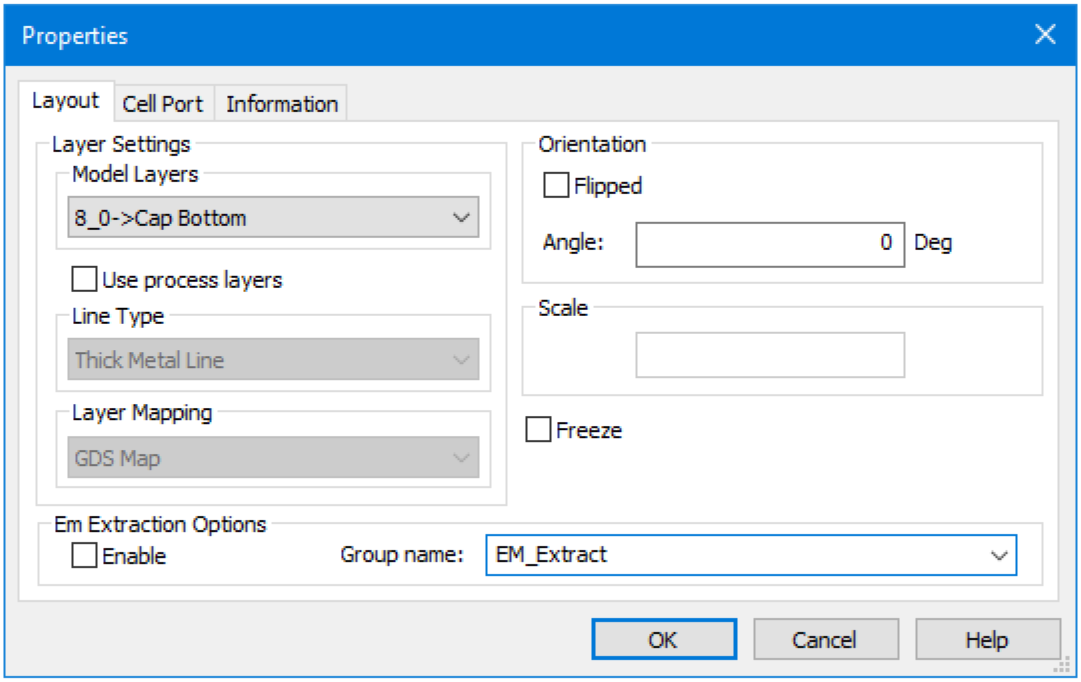
In the Model Layer Mapping, add the new "#Capacitor top" and "#Capacitor bottom" model layers in the following format and order, and map them to the correct **Drawing Layer**.



C.25. Port_Number: Detached Face(s) on Drawing Layer Drawing_Layer_Name

This error indicates that the cell port is not touching any metal inside the artwork cell. See "Connectivity Checking " for more information.

Check the artwork cell to make sure that the face is at a correct location. Sometimes the port seems attached to the metal but is on a different drawing layer than the metal to which it attaches. To check which drawing layer the port is using, select the cell port, right-click and choose **Shape Properties** to display the Properties dialog box. On the **Layout** tab, check the model layer specified in **Model Layers**.

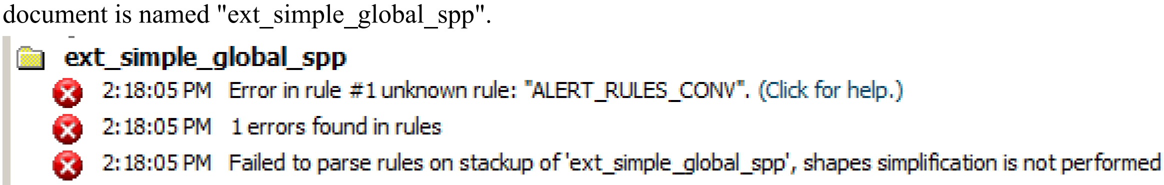


C.26. ALERT_RULES_CONV Error for Geometry Simplification Rules

Starting in AWR Design Environment V10 software, the Shape PreProcessing (SPP) options for AXIEM software are replaced with geometry simplification rules. While the SPP options worked, they were too general because the settings applied to all layers in an EM structure. The new approach of writing simplification rules gives you significantly more power to simplify AXIEM structures.

Due to this change, the use model is slightly different. Previously, SPP options for extraction were set on the EXTRACT block. Now the simplification rules are defined on the Element Options: STACKUP dialog box **Rules** tab. When opening older projects, the old SPP options are converted to an equivalent set of simplification rules. For stand-alone EM structures the conversion is simple-- the new rules are added to that EM document. For extraction, many EXTRACT blocks with different mesh settings can use the same STACKUP, so the conversion from the old SPP options to the new rules can be ambiguous. If a situation is detected that indicates you should inspect the result of the conversion process, an error is purposely introduced so the simulation cannot occur until you fix the rules.

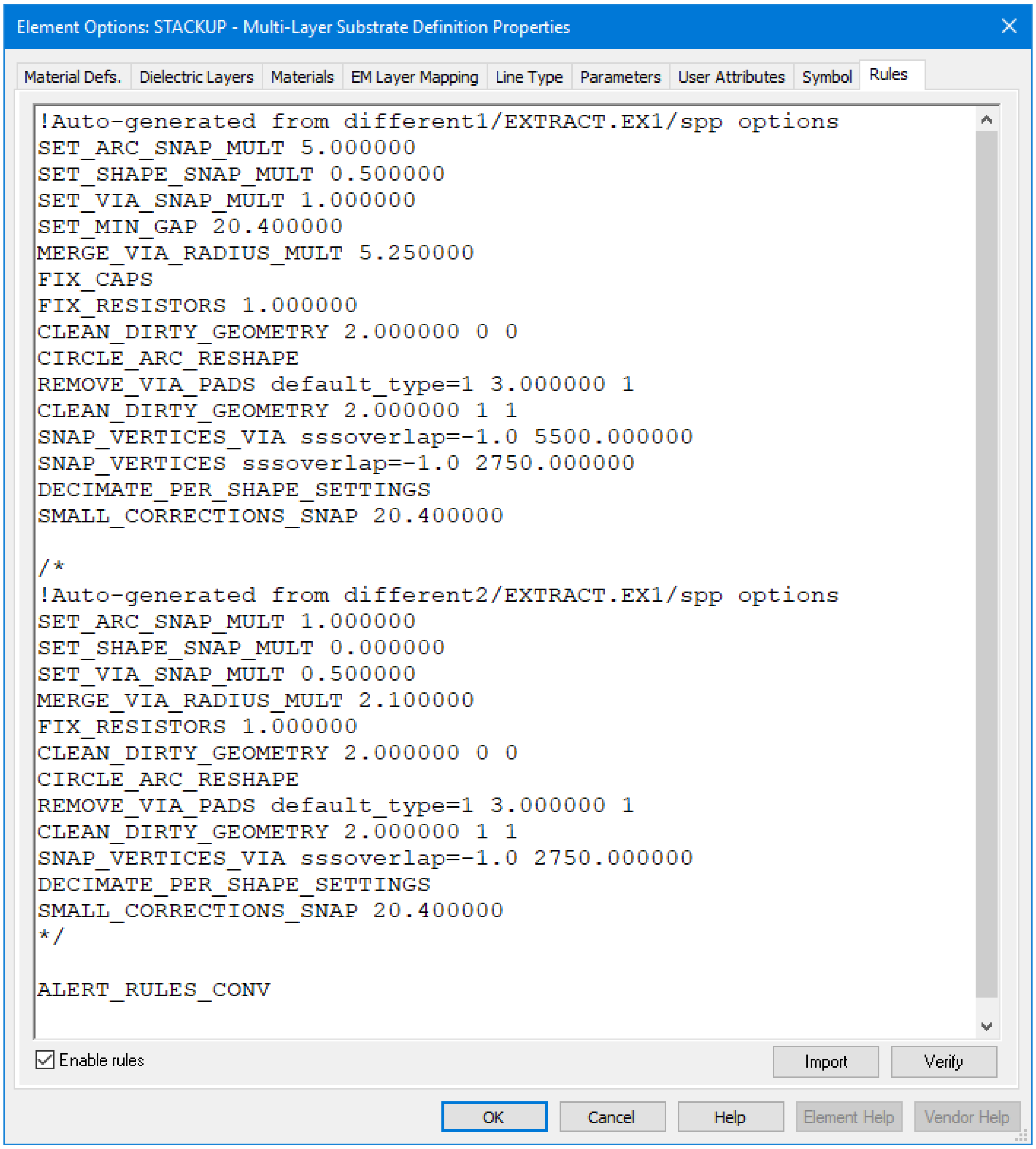
When this specific error displays, you may need to inspect the auto-converted rules. The error message title or the last error line listed indicates which EM document is having the problem. In the following example, the problematic EM document is named "ext_simple_global_spp".



You need to determine which schematic generated this EM extraction document and you need to identify the specific EXTRACT block. With that information you can determine which STACKUP block the EXTRACT block uses. If this is not clear, contact [Technical Support for AWR Products](). After you identify the STACKUP, double-click and view the **Rules** tab to see the automatically generated rules that are causing the issues. The following are several possible cases of multiple EXTRACT blocks using the same STACKUP.

C.26.1. EXTRACT Blocks with Different SPP Options

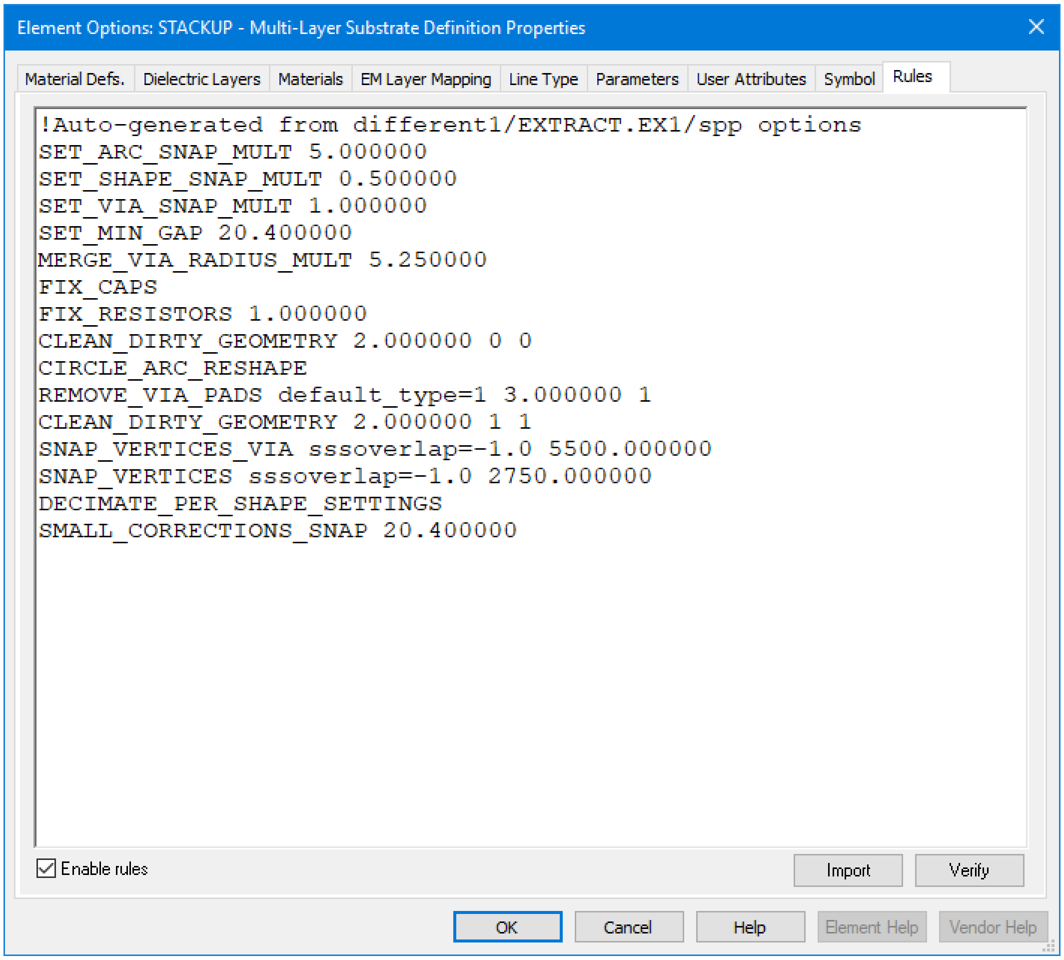
Using EXTRACT blocks with different SPP options settings and pointing to the same STACKUP produces rules similar to the following:



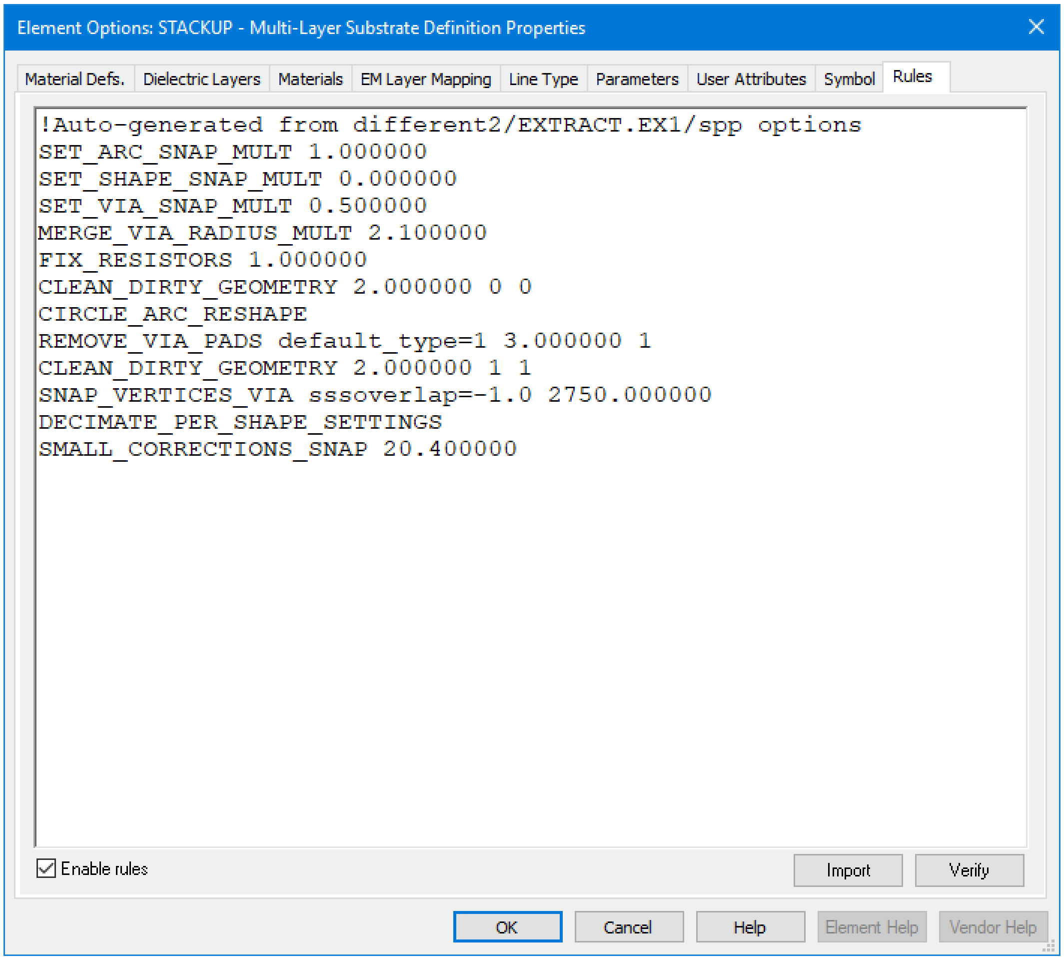
The first line is a comment indicating from which schematic and EXTRACT block the rules come. The first EXTRACT block found writes in the rules. Subsequent EXTRACT blocks then write that block's rules, but they are commented out, and finally the `ALERT_RULES_CONV` rule is written that produces an error.

In this case, the rule sets are different. To clean this up correctly:

1. Make a copy of this STACKUP.
2. In the original STACKUP, delete the commented out rules and the `ALERT_RULES_CONV` rule. The rules should display as:



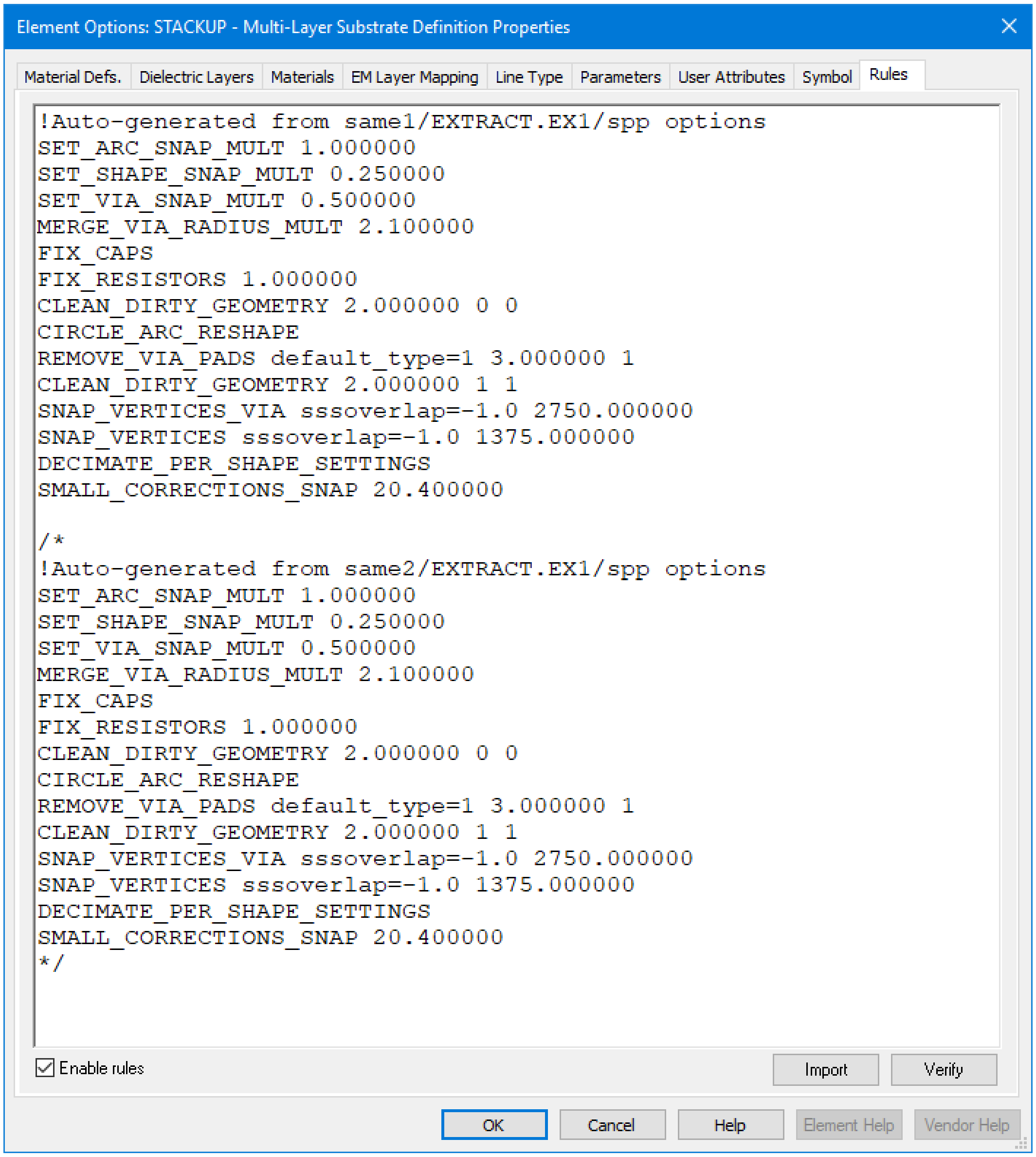
3. In the copied STACKUP, delete the first set of rules, uncomment the second set of rules, and delete the `ALERT_RULES_CONV` rule. The rules should display as:



4. Change the second EXTRACT block (referenced by the second set of rules in this new STACKUP) to use this new STACKUP, instead of the original STACKUP.

C.26.2. EXTRACT Blocks with Same SPP Options

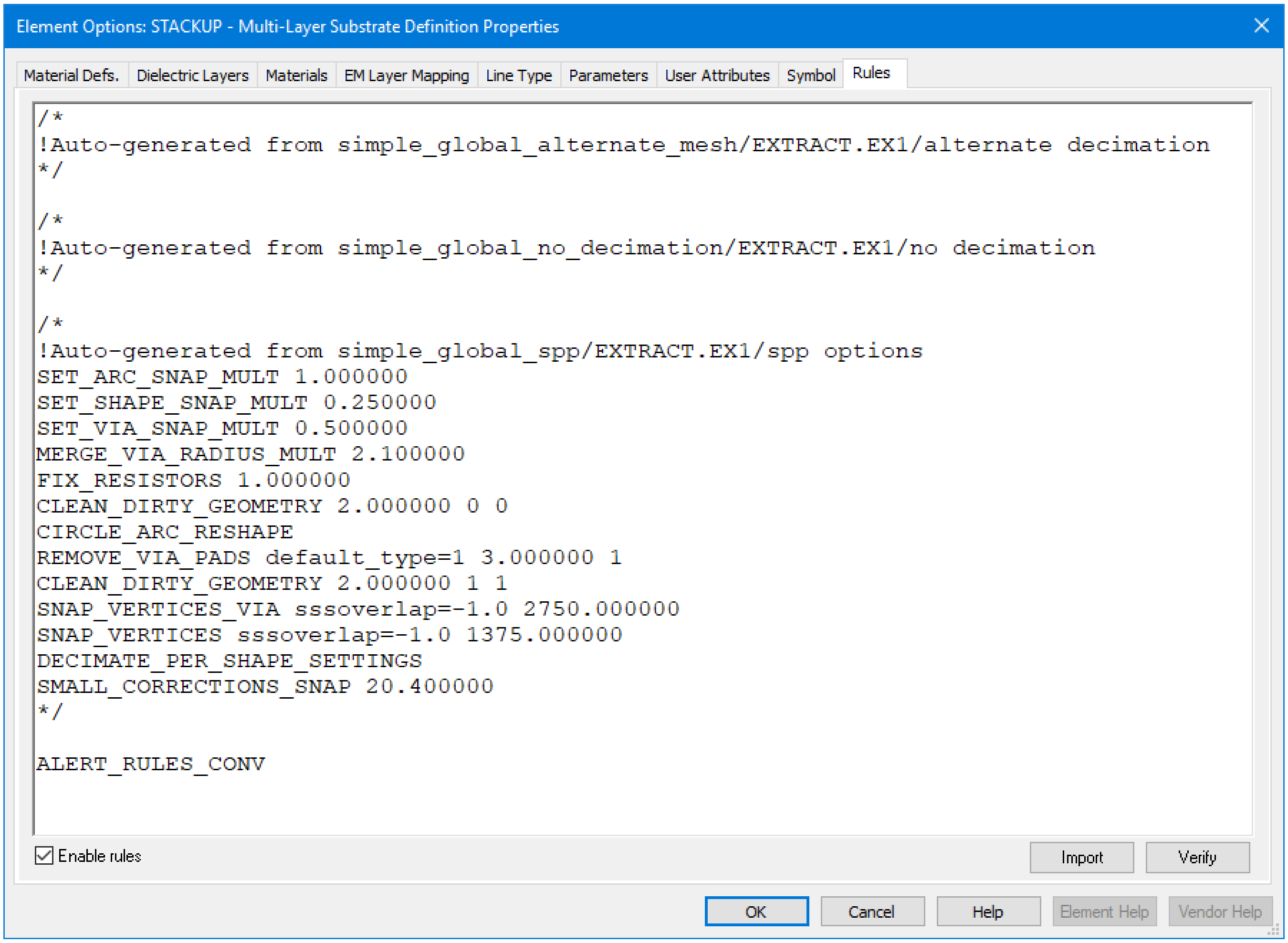
Using EXTRACT blocks with the same SPP options settings and pointing to the same STACKUP produces rules similar to the following.



In this case, the rule sets are the same, and no error is generated. To clean this up correctly, delete the commented out rules.

C.26.3. EXTRACT Blocks with Mixed Mesh Options

Using EXTRACT blocks with a mix of mesh options settings and pointing to the same STACKUP produces rules similar to the following.



A comment line for each EXTRACT block indicates whether the block used "alternate decimation", "no decimation" or "spp options" for mesh options. To clean this up correctly:

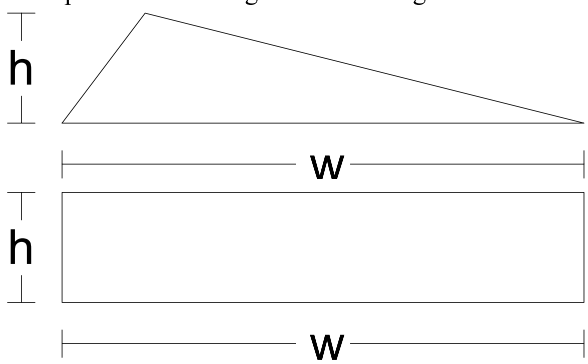
1. Make a copy of this STACKUP.
2. In the original STACKUP, delete the `ALERT_RULES_CONV` rule and uncomment the set of rules.
3. In the copied STACKUP, delete all the rules or clear the **Enable rules** check box.
4. Change the EXTRACT blocks referenced by the "no decimation" or "alternate decimation" comment lines in the original STACKUP to use this new STACKUP, instead of the original STACKUP.

C.27. Shape Modifier Priority Ordering Conflict Detected

You can define the order in which shape modifiers are executed. Different types of modifiers should run before others. If an order issue is detected, this warning displays. For information on proper shape modifier order, see "Layout Modifier Order".

C.28. AXIEM High Aspect Ratio Facet Detected

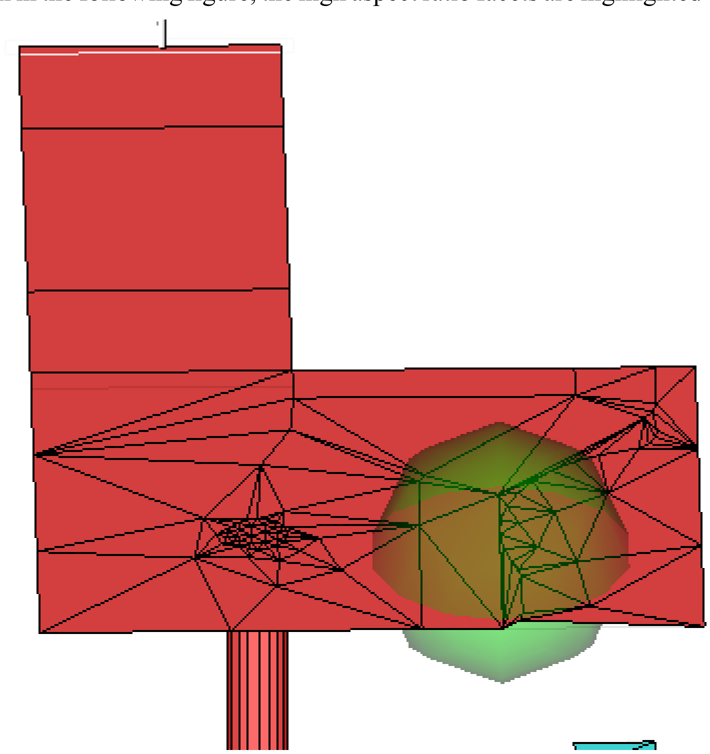
When AXIEM software creates the mesh to simulate, each mesh area is called a facet. A facet can be either a rectangle or a triangle. Each facet has an aspect ratio that is the ratio of the width (w) to the height (h) of the face. The following examples show a triangle and a rectangle.



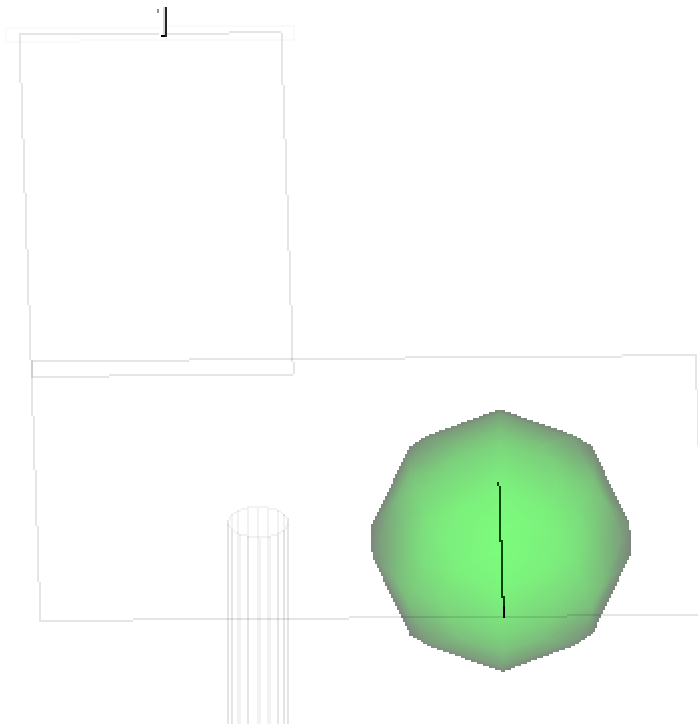
The aspect ratio is calculated for each facet; any facet with a ratio greater than 100 results in a warning, and any facet greater than 1000 results in a warning and does not simulate.

When the Status Window reports a high aspect ratio or warning, you can view these by following these steps:

1. Right-click the EM document in the Project Browser and choose **Add Annotation**.
2. In the Add EM Structure Annotation dialog box, select **Planar EM** as the **Measurement Type** and **HI_ASPECT_FACETS** as the **Measurement**, and then click **OK**.
3. Open the 3D layout of the EM structure.
4. As shown in the following figure, the high aspect ratio facets are highlighted with green spheres to show their location.



As shown in the following figure, you can also turn off the mesh annotation to display only the facets that are high aspect ratio.



High aspect ratio facets are a sign that the mesh is not appropriate to solve the problem. They unnecessarily increase the number of facets, and they can also cause numerical instabilities when solving the problem.

High aspect ratio facets are created when the shapes sent to AXIEM software cannot be properly meshed without creating them. Common causes are:

- Shapes that are intended to be perfectly aligned, but are not, and have a small offset in size. The offsets from plated lines in a MMIC process are usually not small enough to cause this.
- Curved structures that the mesher is not directed to simplify.

The following are options for removing high aspect ratio facets:

- Edit the geometry directly to clean up areas where the facets occur. This can be difficult for complex geometries.
- Allow the mesher to slightly alter the geometry to achieve a good mesh. On the **Mesh** tab of the EM Options or Options dialog box, for **Final Cleanup Options**, **Auto** is recommended and is a good setting to check.
- Write geometry simplification rules to clean up the geometry sent to AXIEM software. See "Geometry Simplification" for more information on geometry simplification and "Generic Procedure for Fixing High Aspect Ratio Facets" for specifics on rules for this specific problem.
- If you use a Process Design Kit (PDK), check with your PDK supplier to see if there are proper geometry simplification rules for the process.

- You can override this error. By default, an aspect ratio of 100 produces a warning and 1000 produces an error and does not simulate. The **HARF Error Threshold** setting on the **Mesh** tab of the EM Options or Options dialog box controls the error limit. Note that this is a secondary option, so you must click the **Show Secondary** button to display it.

C.29. AXIEM High Aspect Area Facet Detected

When AXIEM software creates the mesh to simulate, each mesh area is called a facet. AXIEM software calculates the maximum facet area and an area ratio defined as AR = sqrt(Amax/Ai). The sqrt is used to normalize the area back to equivalence with an edge length so it can be compared to the facet aspect ratio. Any facet with a ratio greater than 100 results in a warning, and any facet greater than 1000 results in a warning and does not simulate.

See "AXIEM High Aspect Ratio Facet Detected" for details on how to detect and fix these facets.

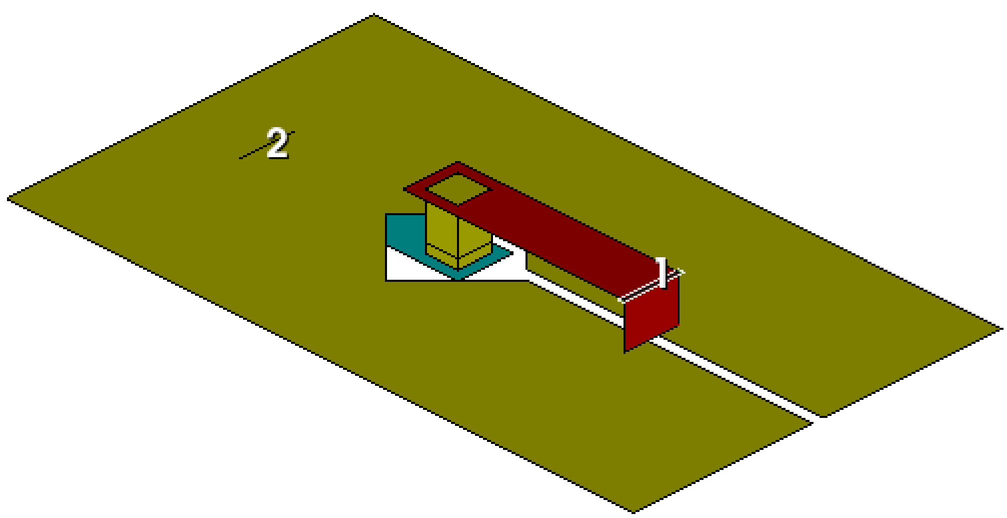
C.30. AXIEM Poor Resolution Facet Detected

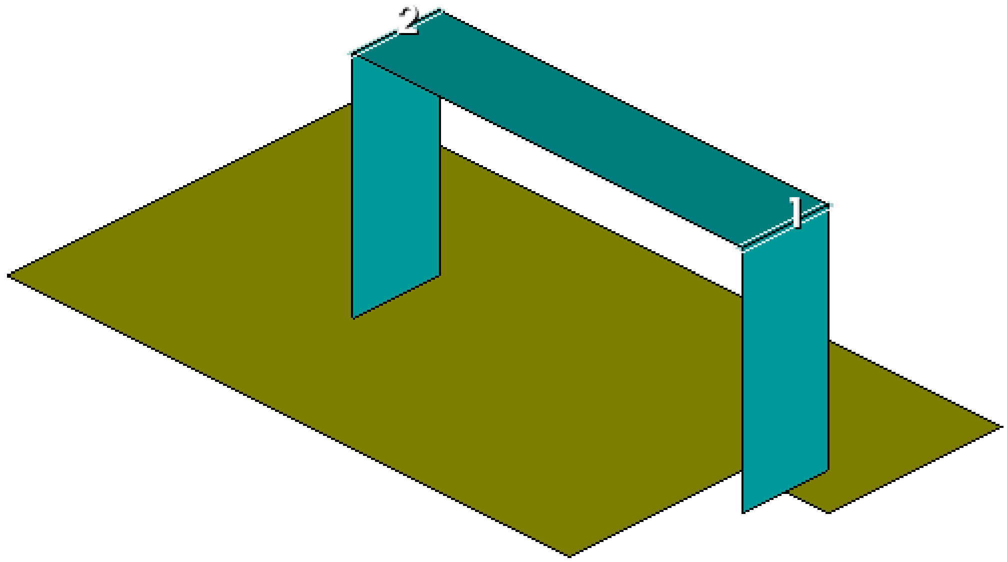
When AXIEM software creates the mesh to simulate, each mesh area is called a facet. This error is a very rare situation in which a facet is not properly created. If this error displays you should contact Cadence technical support.

C.31. AXIEM Local Ground does not Extend Entire Width of the Port Extension

When using a port with explicit ground reference and a local ground plane, the ground plane must be continuous across the entire width of port. There must not be any gaps or cut-outs in the ground plane where the ground plane intersects the vertical strip representing the explicit ground reference.

In the following figures, Port 1 shows invalid ground plane setups, which result in the following error: "Extension down does not extend to the same local ground across the entire width of the port".





C.32. AXIEM Min Edge Length Warning and Port Width Error

Improper AXIEM software mesh settings can cause similar messages to these to display in the Status Window:

- The specified Min Edge Length for the mesh is too large when compared to wavelength at the meshing frequency (8.0GHz). Lower the minimum edge length by 10% to correct this issue.
- The Max Edge Length for the mesh (Min Edge Length x Max Aspect Ratio) is too large when compared to wavelength @ the meshing frequency (6.0 GHz). The Max Edge Length will be limited to 24% of its specified value. Maintaining the specified Min Edge Length will result in an effective Max Aspect Ratio of 2.5.
- Port# 1's width is too small compared to min edge length specified(1.25mm). Either adjust port width or mesh options so that port width is greater than 25% of min edge length.

Note that frequency, lengths, and other values shown in the above messages are design dependent, and may vary from the examples given.

In most cases, the AXIEM software mesh option **Min Edge Length** needs to be reduced in order to resolve the issue. A smaller value for **Min Edge Length** also results in a larger mesh count, and possibly longer simulation time.

The **Min Edge Length** value determines the minimum edge length of a mesh cell. The maximum edge length of a mesh cell is determined by multiplying the **Min Edge Length** by the value of the **Max Aspect Ratio** mesh option.

The meshing frequency also limits on the maximum edge length. The upper limit for maximum edge length is set as a fraction of the meshing frequency wavelength. The fraction depends on the **Mesh Density** setting. When the maximum edge length calculated from (**Min Edge Length**x**Max Aspect Ratio**) exceeds the upper limit determined by the mesh frequency, the first warning above is issued, and the maximum edge length is automatically reduced, resulting in a smaller effective **Max Aspect Ratio** than specified. To maintain the specified **Max Aspect Ratio**, you need to reduce the **Min Edge Length** value.

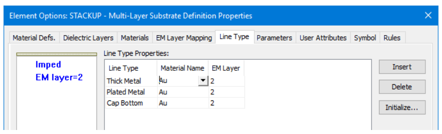
Port widths must also be at least 0.25 (**Min Edge Length**). If a port width is too small compared to **Min Edge Length**, the third error above is issued. To fix the error, either increase the port width, or reduce **Min Edge Length**. Increasing the port width fixes the error without increasing the mesh count.

If the mesh option **Mesh Units** is set to "Relative to Grid", then you should also review the Enclosure **Grid_X** and **Grid_Y** values, since **Min Edge Length** is defined relative to those values. Reducing the Enclosure grid size also reduces the absolute value of **Min Edge Length**.

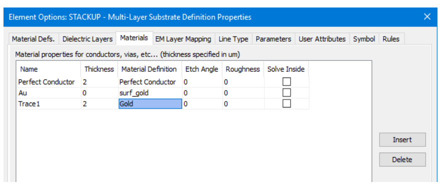
C.33. ACE Simulation when Using Metal Surface Impedances

When you set up a STACKUP block to simulate with ACE, you cannot map any of the line types to a material that is defined with a surface impedance. ACE needs to resolve each model to a substrate, and the substrates have parameters for metal conductivity and metal thickness. A metal surface impedance cannot be uniquely resolved to a conductance and thickness.

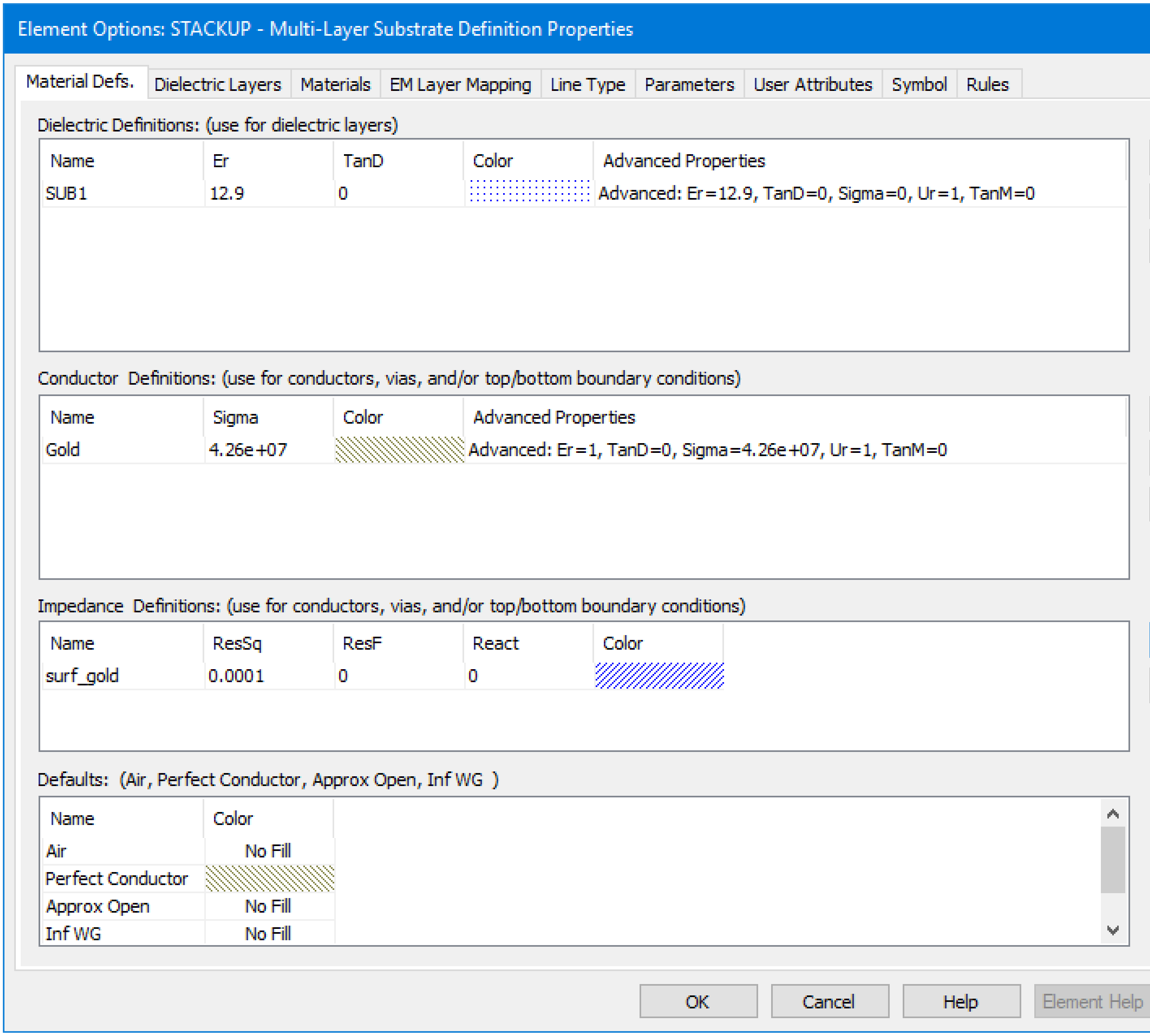
To check these settings, on the schematic, double-click the STACKUP used for your ACE extraction, and then click the **Line Type** tab on the Element Options dialog box. The following figures show an example of this problem.



Each line type has a **Material Name** setting. You can also see in the cross-sectional diagram on the left the type of material used, in this case "Imped". Note that the line types are mapped to the material "Au". You can click the **Materials** tab to find this material.



The material name for "Au" is assigned the material definition of "surf_gold". You can click the **Material Defs** tab to find this bulk material definition.



You can see that "surf_gold" is defined as an impedance with a resistance per square.

In this example, setting the mapping for each line type to use the material "Trace1" (which is defined as the conductor "Gold" on the **Material Defs** tab, with a thickness set to 2um set on the **Materials** tab) is the correct answer.

C.34. Error Obtaining the Antenna Data

There are two likely causes for this warning message:

- Antenna calculations are not yet performed, so the antenna data does not exist. Calculations for antenna measurements and annotations are only performed when needed. To plot antenna measurements or annotations, you must add the measurements or annotations before simulation occurs. Otherwise, the antenna data does not exist in the data set, and an additional simulation is needed. To resimulate after adding antenna measurements, right-click the EM Structure and choose **Force Re-simulation**, then simulate the structure again.

- The antenna was simulated with AFS enabled. Because only a subset of frequencies are simulated with AFS, antenna data is only available for those simulated frequencies, and not the entire frequency list. In this case, you must use FSAMP to specify the frequency sweep of the measurement or annotation. See “Plotting Currents and Antenna Measurements with AFS ” for more information on using AFS and FSAMP. Otherwise, a resimulation with AFS disabled is required to plot antenna measurements at all frequencies.

C.35. Error Reading Image Data

This error displays when you try to plot EM currents when the current data is unavailable. Several situations can cause this error.

Normally, current data is not calculated unless there is an enabled current annotation at the time of simulation. If the annotation is added after the EM structure is simulated, a resimulation is required to generate the current data. To resimulate after adding a current annotation, right-click the EM Structure and choose **Force Re-simulation**, then simulate the structure again. To calculate currents for every simulation, on the “Options Dialog Box: General Tab” dialog box under **Save Results in Document**, select **Currents** .

Current data may also be unavailable if the EM structure is simulated with AFS enabled. Because only a subset of frequencies are simulated with AFS, currents are only available for those simulated frequencies, and not the entire frequency list. To plot EM currents with AFS, you must use FSAMP to specify the frequency sweep of the measurement or annotation. See “Plotting Currents and Antenna Measurements with AFS ” and “In-Situ Current Animation” for more information on using AFS and FSAMP. Otherwise, a resimulation with AFS disabled is required to plot currents at all frequencies.

Current data is also unavailable if the simulator does not support the current annotation. For example, Cadence Analyst™ 3D FEM EM simulation does not support the “EM Document Circuit Current: EM_CKT_CURR” annotation, which generates this error.

C.36. Singular Matrix in Sparse Circuit Solver

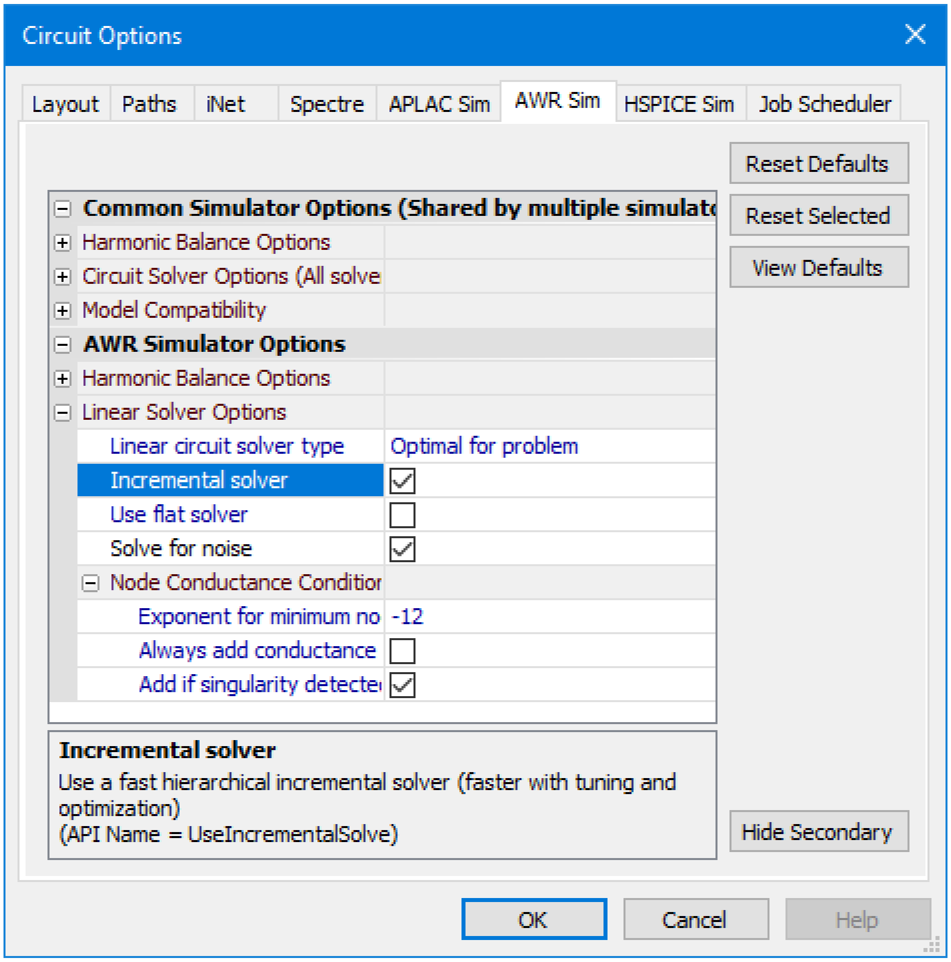
This error is caused by elements in schematics that are not connected or are floating. You should scan the schematic for any elements that have "X"s on the connections instead of solder dots. This is an indication that the part is not connected. To fix this problem, use one of the following solutions:

- Connect all pins of the elements in the simulated schematic.
- Delete or disable (right-click the element and choose **Toggle Enable**) the elements that have no connectivity to your circuit.
- Choose **Options > Default Circuit Options** to display the Circuit Options dialog box. Click the **AWR Sim** tab and under Linear Solver Options, change **Linear circuit solver type** from **Optimal for problem** to **Full Solver**.

C.37. Linear Simulation Error About Y-Matrix

During circuit tuning or optimization, if a "Failed to factor Y matrix" error displays, try increasing **Minimum node conductance** (sweep point: Freq=0 GHz)," or similarly when an "Error during linear simulation of (schematic name). Failed to create reduced Y-matrix in linear solver." error displays.

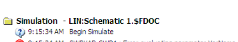
When tuning and optimizing with the AWR Design Environment platform, the simulators organize the matrices that must be solved so as to make the simulation as efficient as possible. In rare cases, this can cause simulation problems; the previous error displays and simulation results are incorrect. To correct this, choose **Options > Default Circuit Options** to display the Circuit Options dialog box, click the **AWR Sim** tab and under Linear Solver Options, clear the **Incremental solver** check box.



This setting makes tuning slower (possibly noticeable on a big circuit) but it eliminates any matrix problems the solver has.

C.38. Error Evaluating Parameter VarName

A "VarName: Attempt to get string on non-string domain" or "Error evaluating parameter VarName" error occurs during simulation while using a swept variable block.



This error is due to a swept variable block setup; the VarName parameter must be enclosed in quotation marks for it to recognize the appropriate variable to sweep.

C.39. Simulating Outside Supported Range of Element

An "(Element Name).(Element ID) : (Parameter) must be (Parameter Restriction)" error message can occur during simulation if you try to simulate a circuit outside of the supported range.

For example, you can use the CPWTEEX element as long as W3/(W12+2*S12) is less than 0.9. Parameter W12 is the conductor width at ports 1 and 2, parameter S12 is the Gap width at ports 1 and 2, and parameter W3 is the conductor width at port 3. If you input values of these parameters such that W3/(W12+2*S12) is greater than 0.9, simulation errors occur.

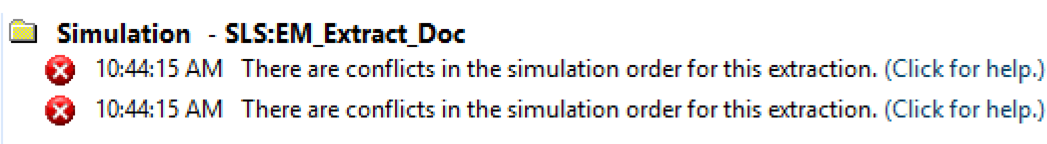
To fix this error, double-click the error message in the Status Window. The schematic containing the element responsible for the error opens with a box around the element. Double-click the element to open the Element Options dialog box, then click the **Parameters** tab and change the value of the parameter(s) causing the problem. Click **OK** to exit the dialog box and then resimulate your circuit. You can view parameter restrictions and recommendations for an element by double-clicking the element in the schematic to open the Element Options dialog box, then clicking the **Element Help** button. In the Help, see the "Parameter Restrictions and Recommendations" section.

C.40. Negative Frequency Folding

Negative frequency folding occurs when working with complex envelope signals whose center frequency is less than one-half the sampling frequency. When this occurs, part of the sampling frequency band contains negative frequencies. Conceptually, negative frequency content is equivalent to the complex conjugate of the corresponding positive frequency content. When the center frequency is greater than 0, the default behavior of VSS spectrum measurements is to automatically convert negative frequency content to the equivalent positive frequency content, or to "fold" the negative frequency content into the positive frequencies.

See Section 3.7 of the *VSS Modeling Guide* for more information on noise and negative frequency folding.

C.41. Conflicts in Simulation Order for Extraction

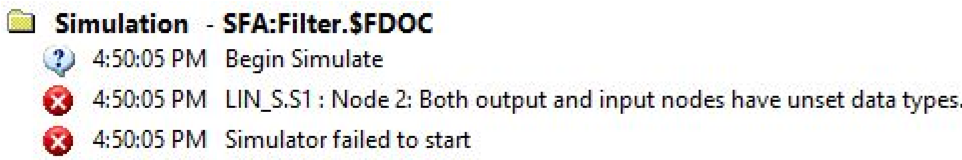


This error is usually seen when a schematic uses a subcircuit that is an EM structure generated through extraction. The error occurs when the simulation of the parent schematic (the schematic with the extracted EM subcircuit), occurs before the simulation of the schematic that does the extraction that creates the EM structure used as the subcircuit.

You can use either of these simple solutions to this problem:

1. If the EM structure represents the schematic that created it (multiple extract groups are not used), then use the schematic as the subcircuit in the top level schematic instead of the EM structure.
2. Disable the EXTRACT block(s) that generate the EM structure(s) used as subcircuits in the top level schematic.

C.42. Unset Node Data Types



This error displays when a data type is not specified in a VSS system diagram.

To set the data type of the of the source(s) in your system diagram, double-click the triangle near the block's node to display the System Node Settings dialog box.

C.43. Doc is Parameterized and Has No Swept Parameters

When an EM structure is parameterized, its use as an EM subcircuit in hierarchy is assumed, so only the top level EM structure simulates, and subcircuits do not simulate. However, if the EM subcircuit contains swept parameters, the EM subcircuit simulates if a measurement or annotation is made on it.

This error occurs when a measurement is a made on an EM subcircuit that is parameterized, but contains no swept parameters. To fix this problem, use one of the following solutions:

- If you do not want measurements on the parameterized EM subcircuit, disable all measurements on the subcircuit.
- If you want measurements on the parameterized EM subcircuit, modify the subcircuit so that it is swept. Add a dummy SWPVAR, and sweep a parameterized variable with only 1 sweep point.

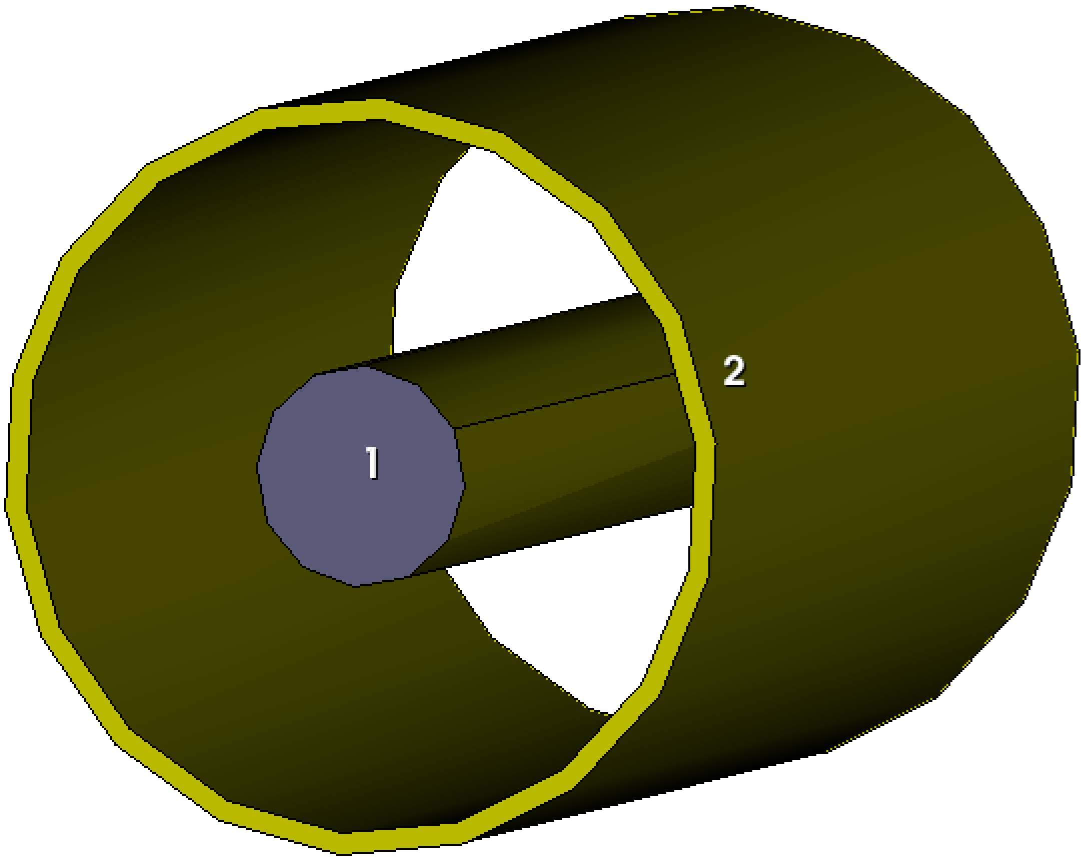
For more information about parameterized EM structures, see "Parameterizing EM Structures" in the *AWR Design Environment Simulation and Analysis Guide*.

C.44. Found Only Good Conductors on Wave Port Plane

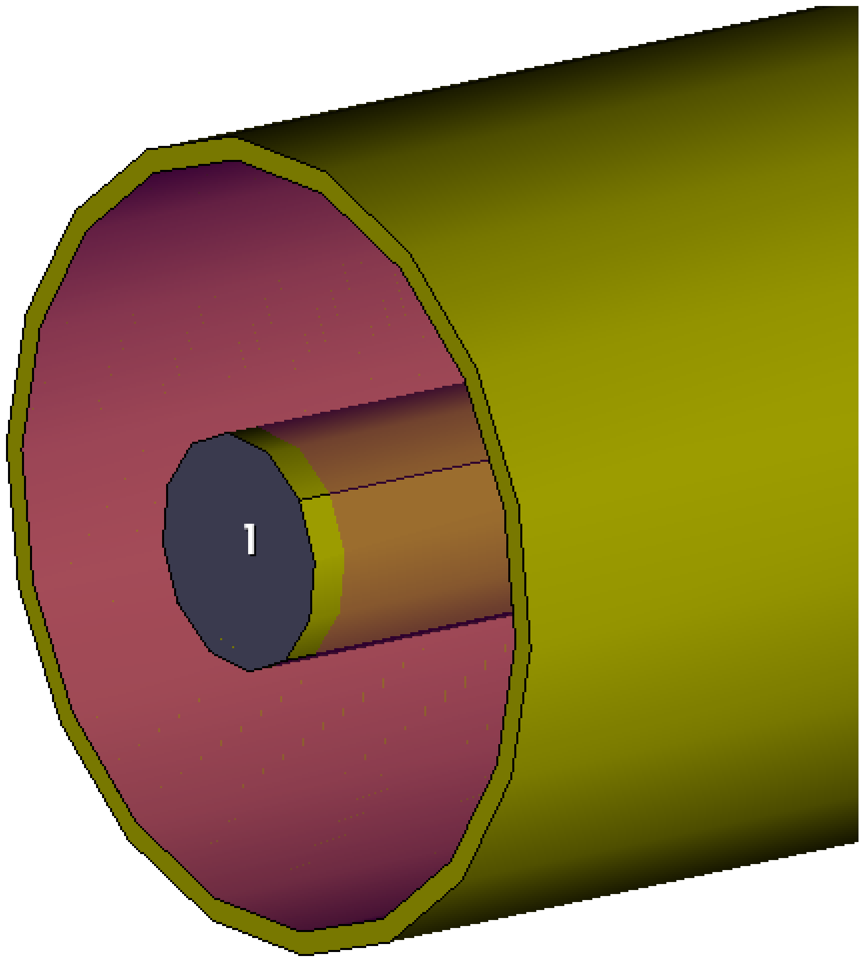
This error occurs when an Analyst wave port is incorrectly defined. Only conductors are identified on the port surface; no dielectrics. Either no dielectric material is defined on the port plane, or the dielectric material is electrically isolated from the conductor. Without the dielectrics, no modes can be defined during the wave port solve.

To fix this error, verify that material properties are applied correctly and that there is no unintended gap between the port face and its adjacent dielectric. Any such gap is treated as an outer PEC boundary and isolates the dielectric from port face if the gap exists outside of a drawn boundary shape. In the Analyst 3D editor, the default boundary condition for unassigned exterior faces is PEC.

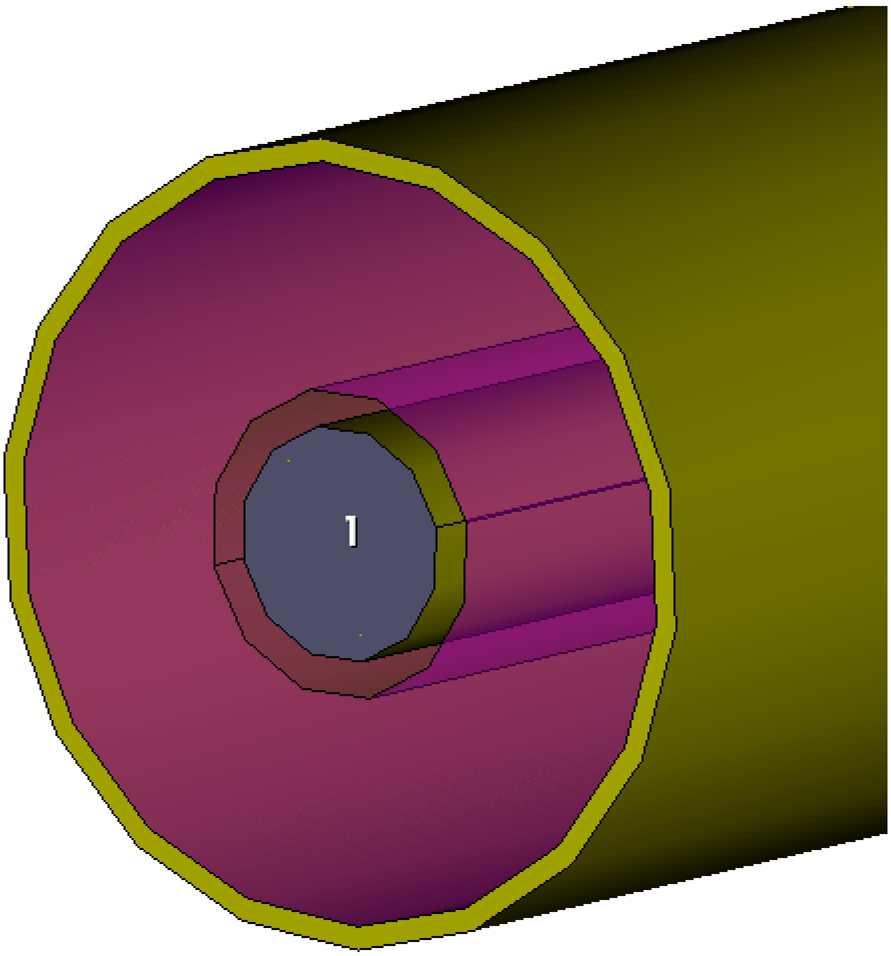
This error can occur with an improperly drawn coaxial line. In the following figures, the coaxial line extends outside of an Analyst boundary shape, therefore the default boundary around the line is PEC. In the example, only the inner and outer conductors are drawn; no insulator is drawn. In this case, there is no dielectric defined, and the space that the insulator would have occupied is assigned the default PEC boundary condition.



In the following figure, the inner conductor of the coax protrudes beyond the insulator. There is no dielectric defined on the plane of the port face, and the insulator is shorted-out by the default PEC boundary condition.



In the final figure, there is a gap between the inner conductor and insulator. This gap is considered an exterior boundary, and treated as the default PEC boundary condition. Again, the inner conductor is isolated from the insulator by the background PEC boundary condition. In this figure, the gap is exaggerated so it is easily seen. It is possible for the gap to be very small and only visible after zooming around the edge of the inner conductor.



For more information about Analyst ports, see "Analyst Ports" in the *AWR Design Environment Simulation and Analysis Guide*.

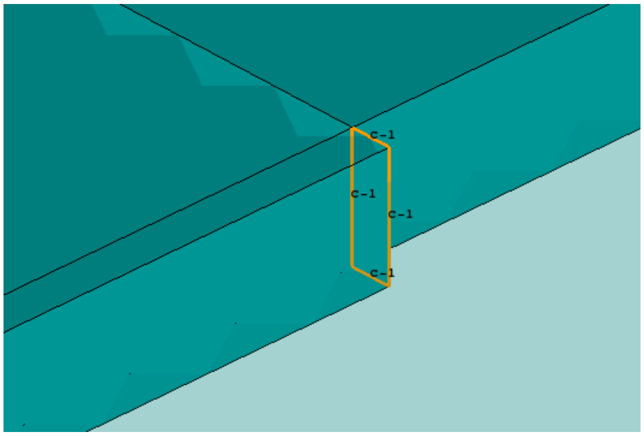
C.45. Analyst Potential Geometry Problem Found

This can be a warning or error message displayed in the Status Window when the Analyst simulator finds any geometry problem that potentially causes a bad quality mesh or non-physical structure for EM simulation.

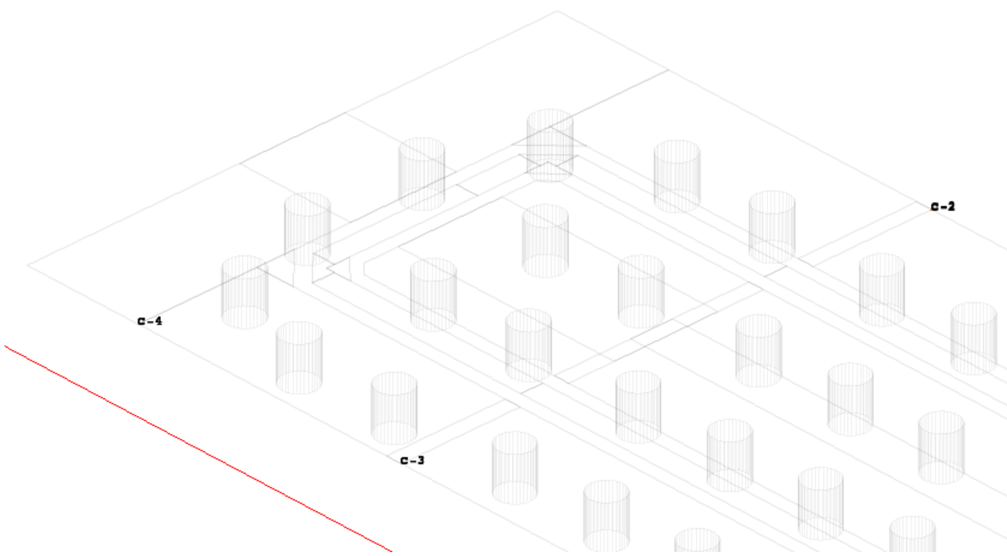
Assuming that you attempt to mesh or simulate the structure, when this warning or error message displays, you can follow these steps to find the location(s) of the geometry problem(s):

1. Right-click the EM document in the Project Browser and choose **Add Annotation**.
2. In the Add EM Structure Annotation dialog box, select **EM 3D** as the **Measurement Type** and **EM3D_SURF_CONCERNS** as the **Measurement**, and then click **OK**.
3. Open the 3D layout view of the EM structure if you have not done so.

4. In the Project Browser under the associated EM structure, find the "MESH_3D" annotation, right-click and disable it.
5. The following figure show the potential geometrical problems highlighted in yellow with label "c-1" to show their locations. In the text label, the letter "c" means concern, and the number "1" is an identifier on the specific occurrence that you can use to cross-reference with the message in the Status Window.

   In some cases, a wireframe view of the geometry gives a better perspective on the location of the problem areas. This is illustrated in the following figure. You can obtain the wireframe view by right-clicking on the "EM3D_SURF_CONCERNS" annotation and choosing **Toggle Enable** twice.

The Analyst solver detects potential geometry problems when performing shape processing, initial meshing and mesh refinement. These problems may lead to low quality mesh, thereby slowing down the convergence of the finite element solvers. Some problems could also produce incorrect physics and cause simulation failure.

Some of the common causes of geometry problems are:

- The 3D imported CAD model is defective.
- Geometrical details such as holes, chamfers, blends that require many tiny mesh elements to represent accurately.
- 3D planar shapes that are intended to be perfectly aligned, but are not, creating small overlap or gap between them.

- Small gap or intersection between 3D planar shapes created in the Microwave Office Layout Editor and true 3D solids created in the AWR 3D editor.

The following are options for removing potential geometry problems:

- Edit the geometry directly to align different shapes or remove small geometrical details that are irrelevant for electromagnetic simulation. Use the face removal tool in the AWR 3D editor to remove small chamfers and fillets.
- Take advantage of the priority rules the Analyst solver uses to determine the material property of each region, and enlarge the volume of certain solids to get rid of small gap or intersections. For example, if a conductor touches a dielectric, you can enlarge the dielectric such that its boundary face lies completely inside the conductor. In this way the model created by Analyst software simulation would have the correct boundary when the Analyst solver applies the Boolean engine to separate the problem domain into regions with different materials.
- Write geometry simplification rules to clean up the 3D planar shapes. See “Geometry Simplification” for more information on geometry simplification.
- Fill any small gaps or intersections between 3D planar shapes created in the Microwave Office Layout Editor and true 3D solids created in the AWR 3D editor by artificially inflating one of the shapes, with care not to impact the design.
- Turn off **Substitute True Arcs** at the EM document level or on individual shapes causing problems.

C.46. Incompatible Data Types

This VSS error occurs when an input or output port with one data type is connected to another port with a different, incompatible data type.

In the VSS program the input and output ports of a block may have one of the following data types: Digital, Fixed-Point, Real, Complex, and Unset. With the exception of Unset, input and output ports may only be connected to ports of the same data type.

See Section 1.4 of the *VSS Modeling Guide* for more information on VSS data types and signals.

C.47. Incompatible Auto Data Types

This VSS error occurs when an input or output port with one data type is connected indirectly through ports with the Unset data type to a port with a different data type.

In the VSS program the input and output ports of a block may have one of the following data types: Digital, Fixed-Point, Real, Complex, and Unset. With the exception of Unset, input and output ports may only be connected to ports of the same data type.

Blocks with input and output ports with the Unset data type typically match the data type seen at one port with the other ports. For example, the RF Attenuator block RFATTEN has both its input and output ports Unset by default. If the input port is then connected to an output port with the Complex data type, the output port is treated at simulation time as if it has the Complex data type. The data type of a port that is by default Unset may also be explicitly set by double-clicking the triangular marker in the block's symbol where the port's line joins the block. The System Node Settings dialog box displays to allow you to change the port's data type. Note that you can only select supported data types. For example, the RFATTEN block only supports Real and Complex data types; Digital and Fixed-Point are not supported so you cannot select them.

See Section 1.4 of the *VSS Modeling Guide* for more information on VSS data types and signals.

C.48. Cannot Take Measurements on System Diagrams with PORTDIN Blocks

This VSS error occurs when you try to run a VSS simulation with a measurement that refers directly to a system diagram that contains a PORTDIN block, and that system diagram is not a subcircuit within another system diagram. If you take a measurement within a subcircuit, make sure the measurement refers first to the top-level system diagram, and not directly to the subcircuit.

If you refer to a system diagram that contains a PORTDIN block, but that instance of the system diagram is not a subcircuit, that system diagram cannot be run because the PORTDIN requires an input signal.

C.49. Simulation Deadlock

This VSS Time Domain simulation error occurs when all blocks are waiting on their upstream blocks for more samples, and no block is generating any more samples. To determine where the samples are being held up, use the SMPBKLOG and SMPCNT annotations. SMPBKLOG displays the number of samples being held at the input ports of blocks, while SMPCNT shows the number of samples that have passed through a port to or from another port.

Look for the downstream-most block with a large SMPBKLOG count at its input ports and a small SMPBKLOG count at the input ports connected to its output ports. This block is typically a block that processes samples in chunks, often a Signal Processing block. Examine the parameters of this block to determine how many samples it requires, then examine the upstream blocks to locate where more samples need to be generated.

In some cases you may be able to eliminate the simulation deadlock by increasing the **Max. Node Data Accumulation** setting on the System Simulator Options dialog box **Advanced** tab.. If your system has more than 8 GB of memory, try increasing this value to 1,000,000, or several times the largest value reported by SMPBKLOG.

Another cause for simulation deadlock is not having a large enough number of delayed samples in the delay block of a feedback loop. In this case try setting the number of samples delayed to the maximum backlog reported. Use the DLY_SMP block rather than the other delay blocks to explicitly control the number of samples to delay.

Remember to account for resampling that may occur when interpreting the numbers of samples backlogged. Resampling occurs with blocks such as RESAMPLER, HOLD, and DECIM.

C.50. Node Properties Not Propagated

This VSS error occurs most often in feedback loops when the delay block in the feedback path is not directly connected to the feedback point. The feedback point is typically an ADD or COMBINER block, but may be any multiple input block.

The solution is to move the delay block (which is required in feedback loops) so its output port is directly connected to the input port of the multiple input block at the feedback point.

This error may also occur with some blocks if an instance of the block does not have a source block upstream. This typically occurs when you place such a block on the system diagram and decide not to use it, but leave it on the system diagram. You should either delete or disable the block instance.

This error may also occur when working with multiple blocks with bi-directional ports connected together, when the simulator does not resolve the bi-directional ports as expected. In this scenario you can use the SIGDIR block to indicate the desired signal direction to the simulator.

See Section 1.5 of the *VSS Modeling Guide* for more information on auto-configuration in VSS software, which uses propagated properties.

C.51. Incompatible Center and Sampling Frequencies

This VSS error occurs when either the center frequency or sampling frequency cannot be automatically set in a source block when the system diagram contains multiple source blocks. This often occurs when the center frequency or sampling frequency is explicitly specified in multiple source blocks, but one of the frequencies is specified incorrectly, often due to failure to account for resampling or center frequency shifts.

Explicitly setting the center frequency or sampling frequency to the appropriate value should be the solution. Remember to account for changes to center frequency due to mixers and frequency multipliers, and to changes in sampling frequency due to resampling. Resampling may occur when interpreting the numbers of samples backlogged. Resampling occurs with blocks such as RESAMPLER, HOLD, and DECIM.

See Section 1.5 in the *VSS Modeling Guide* for more information on auto-configuration, Section 1.4.3 for more information on center frequencies, and Section 1.4.4 for more information on sampling frequencies in VSS software.

C.52. Disconnected Elements Causing Ill-Conditioned Matrix

Elements that are disconnected from the circuit but still enabled can unnecessarily slow simulations. These disconnected elements are still included in the matrix solve for the circuit and can cause the matrix to be ill-conditioned, making the solve more computationally expensive.

To fix this issue you should disable the disconnected/isolated elements that have no effect on circuit performance. To disable an element, right-click the element and choose **Toggle Enable**. To disable multiple elements hold down the **Shift** key while selecting them. With multiple elements selected, right-click one of them and choose **Toggle Enable** to disable (toggle Off) all selected elements.

C.53. Missing Element Definition for '<model>'

This error occurs when a project is saved with an element model that is not available in the current session. This can occur if a project includes a third-party model.

The solution is typically one of the following:

- Ensure that you have the necessary vendor-provided models or process development kits (PDKs).
- If the vendor's installation instructions direct you to install their models in the *Models* folder under the AWR Design Environment software installation folder, the instructions are outdated and incorrect for version 11. To create the required folder, choose **Help > Show Files/Directories**, then double-click the *Appdatauser* folder. If necessary, create a folder named *Models* under that directory, and install the models there.
- If the vendor does not provide 64-bit models, you must use the "mixed" version or 32-bit version of the AWR Design Environment software.

C.54. Could Not Determine VSS Node Type

This VSS error occurs when working with VSS blocks that support bi-directional ports. The error occurs when the simulator cannot determine whether a bi-directional port should be treated as an input port or an output port. In many cases this error can be fixed by connecting a Signal Direction SIGDIR block to the bi-directional port that caused the error and configuring the signal direction appropriately.

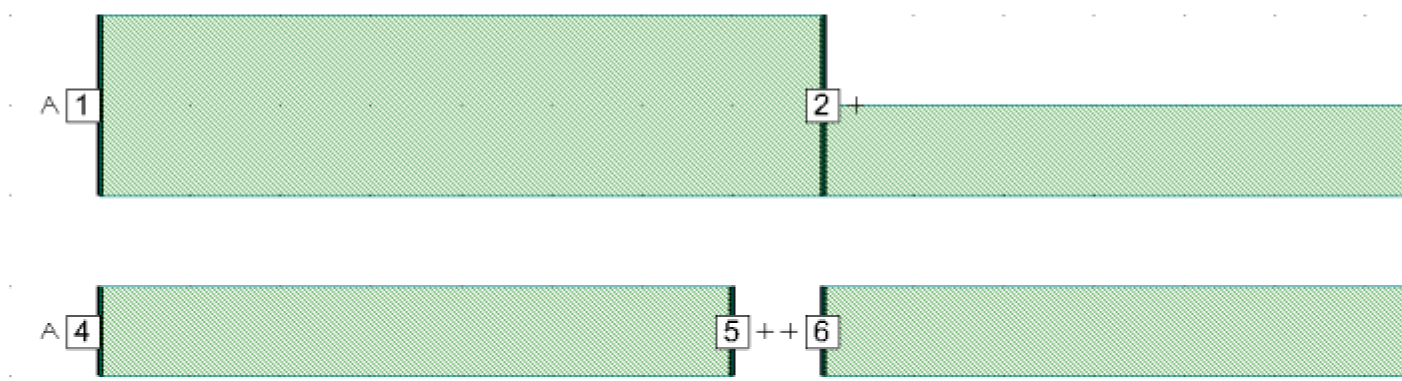
Note, however, that some of the VSS blocks that support more than two bi-directional ports have constraints on how the ports may be configured. Among these blocks are the RF Switch blocks RFSW_1nDYN, RFSW_1nST, RFSW_n1DYN, RFSW_n1ST, and the Splitter SPLITTER block. The limitation is that one port is of one type, while the remaining ports must be of the opposite type. For example, if port 1 of RFSW_1nST is connected to an output port, port 1 is treated as an input port. The remaining ports of RFSW_1nST are then treated as output ports.

For the RF Switch case, if different switched port types are desired, the RF Direction Switch RFSW_DIR should be used. This block has one switched port as an input port, a second switched port as an output port, and the common port dynamically changing depending upon whether the 'on' port is the input port or the output port. This block is ideal for use in TX/RX modules, where the switched input port is connected to the transmit side, the switched output port is connected to the receive side, and the common port is connected to the antenna.

C.55. AXIEM Internal Port Setup Issue

An internal port sets up a 1V potential drop across its face and so requires metal on both sides of the port matching the width of the port.

If one of the metals is wider than the other or if there is metal on only one side of the internal port, the following error message displays: "An internal port must have metal on both sides of the entire extent of the port". In the following figure, internal Ports 2, 5 and 6 are invalid and result in this error.



Verify that the traces on both sides of an internal port are exactly the same width.

Otherwise, you can use an edge port if the port assignment is desired on a trace that does not make contact with another equivalent conductor.

C.56. Analyst Effective Radiation Boundary does not Enclose Radiator

This warning is issued when the radiation boundary condition, PML or Approx Open, is not a fully closed surface.

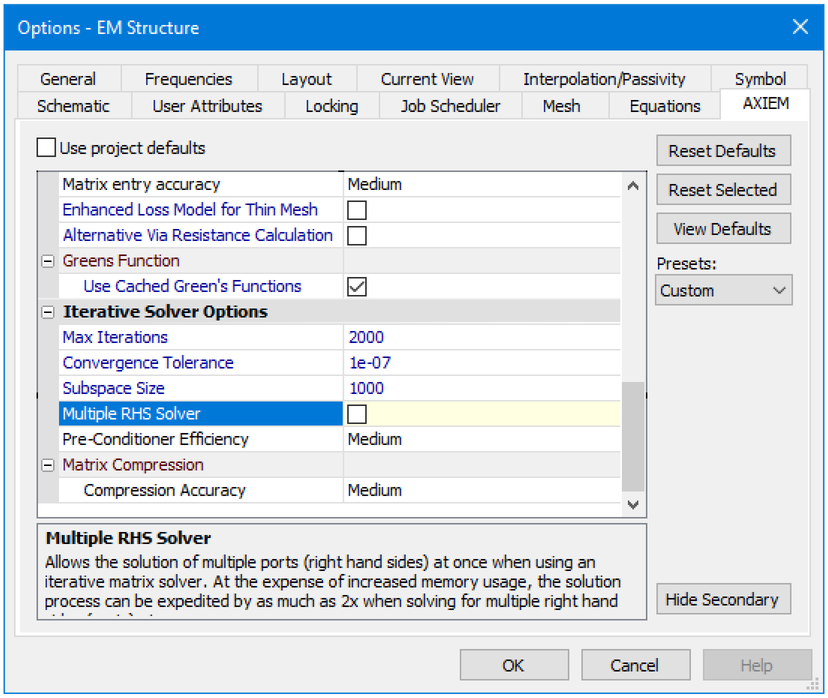
For the best accuracy in antenna far field calculations, the radiation boundary condition should be a closed surface with no breaks, except for intersections with PEC ground planes or symmetry planes. This warning is issued any time the Analyst™ solver detects another structure on the boundary.

Some geometries can be accurately defined with structures or ports on the simulation boundary, as long as only very low fields are radiated in the direction of the boundary structure. Use great care when defining such structures, because they may invalidate the far field calculation if they interfere with the radiated fields too much. For instance, rather than define wave ports on the simulation boundary you get better results with internal wave ports.

C.57. AXIEM Multiple-port Solver Out of Memory

The AXIEM Iterative solver attempts to do simultaneous port solves, which decreases the simulation time at the expense of increased memory usage. If the simulation runs out of memory, an error is issued.

You can disable the parallel port solves by clearing the secondary Iterative solver option **Multiple RHS Solver**.



With this option cleared, the solve time increases because each port solve is now done separately, but the amount of memory needed is also decreased.

C.58. Unsupported Model

In some rare circumstances, electrical models are not supported in all the simulators available in the software. In this case, you can try a different simulator or view the Help for the model to look for information about which simulators are supported. If you are not already using the APLAC simulator, that is the first simulator to try.

C.59. No Connectivity Checking when Using Shape Modifiers

The Connectivity Checker will not run when a shape modifier is used at any level of hierarchy in the design. To work around this problem, disable any shape modifiers and rerun the Connectivity Checker. This error occurs because the Connectivity Checker looks at each level of hierarchy separately, one of which contains the unmodified geometry. When you view the layout for an overall design, however, any lower level that has a modifier shows the modified geometry in the layout. In this case the top level layout might not match the layout from the lower levels and produces false results.

C.60. Global Definition Document '<global doc>' Not Found

This message occurs because the source document (for example; schematic, system diagram, or EM structure) is referencing a missing Global Definitions document. In the Status Window, the message group title contains the source document name, and '<global doc>' is the name of the missing document.

The missing Global Definition document may have been deleted, or not imported along with the source document. In the Project Browser, right-click the source document and choose **Options**, then choose an available Global Definitions document on the **Equations** tab. Otherwise, create or import the required Global Definitions document.

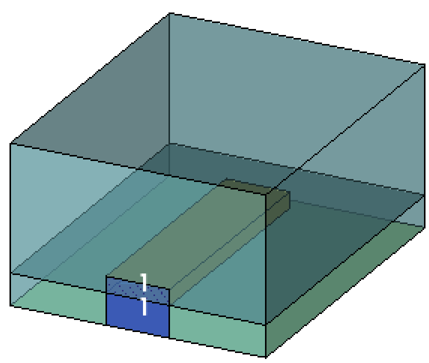
C.61. Illegal measurement component for Loop Gain

The LoopGain measurement only works with specific measurement components: the nonlinear trans-admittance branches of transistors. This allows the measurement to work without breaking the loop, while all connections are intact.

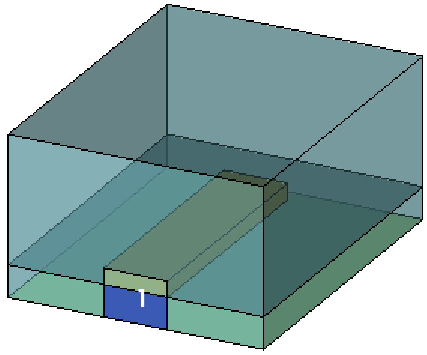
C.62. Analyst Port Polarity Not Defined

This warning is issued when the user has defined a lumped port in the 3D editor without establishing the port polarity. This forces the solver to choose a polarity, which may not be what the user intends.

The polarity of a lumped port can be established in one of two ways. Differential ports have set polarity by definition. When you use a patch, you can force the trace end of the patch to positive polarity by assigning the port attribute to both the patch and the end of the trace. When the port attribute is assigned in this way, the Analyst solver orients the port polarity so that the trace is the positive terminal. If no polarity is indicated by the geometry, the Analyst solver assigns the polarity based upon an analysis of the geometry. If the polarity is important, you should assign it yourself. The lumped port in the following figure is defined with the port attribute on both the patch and the end of the trace, so this port has polarity with the positive end on the trace.



The following figure shows the same system, where the port attribute has not been assigned to the end of the trace. In this case, the polarity is defined by the solver.



C.63. Wave Impedance Invalid

The wave impedance of a propagating mode is defined only in the strictly homogeneous case (so it only exists for well-described modes in homogeneous waveguides) If your simulation contains wave ports on traces or inhomogeneous waveguides, it is not possible to calculate the wave impedance for all wave ports. Instead, the Analyst solver reverts to using the default characteristic impedance method, which is the Power Current method. That characteristic impedance method is used for all wave ports in the simulation, including any homogeneous waveguides.

The characteristic impedance calculation is only performed on wave ports because circuit ports have predefined characteristic impedance given by the impedance in the port definition. A simulation that contains both homogeneous waveguides and circuit ports is able to use the wave impedance method.

C.64. Schematic Symbols Off Grid

The Cadence® Microwave Office® software schematic symbol grid requires that symbol pins fall on a multiple of 100 dbu. The grid is not configurable. Unified Library symbols are scaled by a static factor of 10 when used in Microwave Office software. The Cadence Allegro® Design Entry HDL (DE-HDL) and Cadence Virtuoso® Schematic Editor software allow you to change the schematic grid settings when creating symbols. To ensure symbol compatibility, symbol pins in DE-HDL and Virtuoso software need to use a dbu of 10. Since symbol pins in DE-HDL and Virtuoso software must be placed on the grid, you should choose a grid setting that ensures the pin x and y locations (in user units) multiplied by the dbuperuu is a multiple of 10.

Definitions

- Pin location: Physical location in user units. Must fall on grid. The grid values control where you can draw pins in the symbols.

  NOTE: Only the symbol pins must fall on the grid; this limitation does not apply to symbol graphics.
- dbuperuu: Integer value that controls the fidelity of symbol grid.

To ensure compatibility, `Pin_location(x,y) * dbuperuu = a multiple of 10` must be true.

The following are examples of compatible and incompatible symbol pin placement.

Compatible DE-HDL symbol:

- DE-HDL default user unit (uu) = 1 inch
- DE-HDL default dbu per user unit (dbuperuu) = 500

- Grid value = 10 dbu (or 0.02 inches)

All pins are placed on increments of 0.02 inches, which is 10 dbu, so when used in Microwave Office software they fall on the 100 grid.

Two-pin symbol example:

Pin#	Pin(x) DE-HDL	Pin(y) DE-HDL	Pin(x) AWR	Pin(y) AWR
1	-0.5" or -250 dbu	0" or 0 dbu	-2500 dbu	0 dbu
2	+0.5" or -250 dbu	0" or 0 dbu	2500 dbu	0 dbu

Incompatible DE-HDL symbol:

- DE-HDL default user unit (uu) = 1 inch
- DE-HDL default dbu per user unit (dbuperuu) = 500
- Grid value = 25 dbu (or 0.05 inches)

All pins are placed on increments of 0.05 inches, which is 25 dbu, so when used in Microwave Office software they fall in increments of 250. The symbol pins could be incompatible, depending on their placement.

Two-pin symbol example:

Pin#	Pin(x) DE-HDL	Pin(y) DE-HDL	Pin(x) AWR	Pin(y) AWR
1	-0.55" or -275 dbu	0" or 0 dbu	-2750 dbu	0 dbu
2	+0.55" or 275 dbu	0" or 0 dbu	2750 dbu	0 dbu

Compatible Virtuoso symbol:

- Virtuoso Schematic Editor default user unit (uu) = 1 inch
- Virtuoso Schematic Editor default dbu per user unit (dbuperuu) = 160
- Grid value = 10 dbu (or 0.0625 inches)

All pins are placed on increments of 0.0625 inches, which is 10 dbu, so when used in Microwave Office software they fall on the 100 grid.

Two-pin symbol example:

Pin#	Pin(x) DE-HDL	Pin(y) DE-HDL	Pin(x) AWR	Pin(y) AWR
1	-0.5" or -80 dbu	0" or 0 dbu	-800 dbu	0 dbu
2	+0.5" or 80 dbu	0" or 0 dbu	800 dbu	0 dbu

Incompatible Virtuoso symbol:

- Virtuoso Schematic Editor default user unit (uu) = 1 inch
- Virtuoso Schematic Editor default dbu per user unit (dbuperuu) = 160
- Grid value = 16 dbu (or 0.1 inches)

All pins are placed on increments of 0.1 inches, which is 16 dbu, so when used in Microwave Office software they fall in increments of 160. The symbol pins would be incompatible.

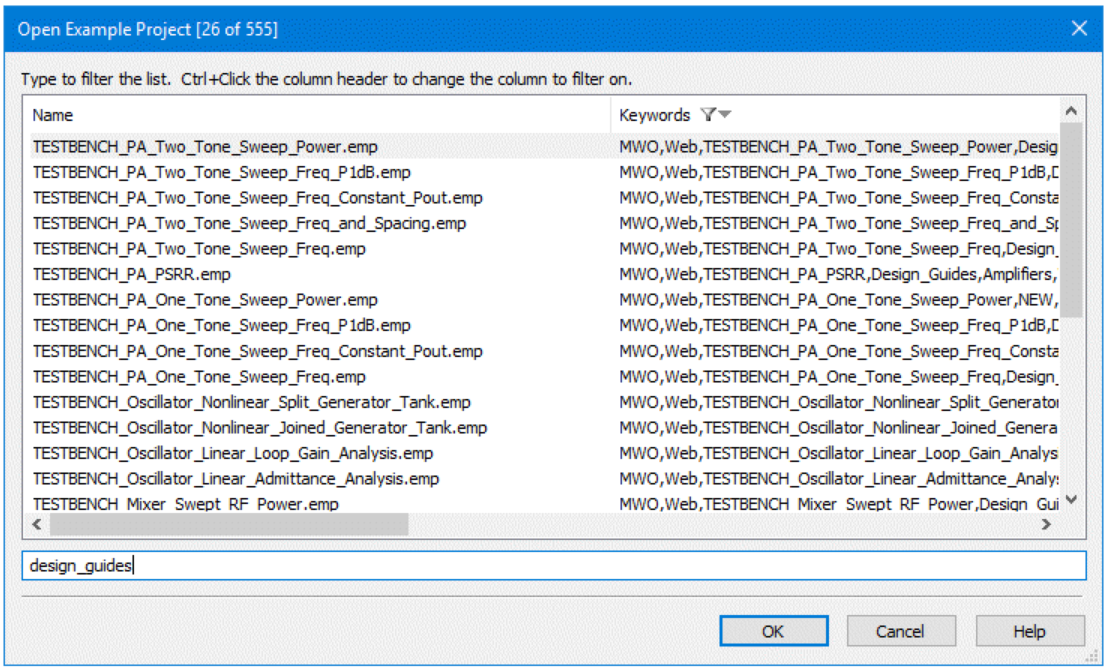
Two-pin symbol example:

Pin#	Pin(x) DE-HDL	Pin(y) DE-HDL	Pin(x) AWR	Pin(y) AWR
1	-0.4" or -64 dbu	0" or 0 dbu	-640 dbu	0 dbu
2	+0.4" or 64 dbu	0" or 0 dbu	640 dbu	0 dbu

Appendix D. AWR Design Environment Test Bench Projects

The Cadence® AWR Design Environment® platform includes an extensive collection of examples. Many of the examples show completed designs that contain different types of analysis. Test bench examples are also available to show a test bench setup and useful measurements taken from that test bench. Test bench examples have the following attributes:

- Example names include "TESTBENCH_", for example, *TESTBENCH_AMP_Dynamic_Load_Line.emp*". Choose **Help > Open Example** to display the Open Example Project dialog box, then type "design_guides" in the text field at the bottom of the dialog box to filter the examples using these keywords, as shown in the following figure.



- These examples are simple projects that focus on a single test bench setup and measurements taken from the test bench.
- Test bench schematics contain notes on what settings you should change.
- Design Notes are minimal. The main test bench schematic has the same name as the project with "TESTBENCH_" removed.
- A schematic using "window in window" is provided to make one view of all of the graphs. The name of this schematic is the same as the main test bench schematic, with "display" appended to it.
- Graphs include text to explain any relevant information about the graph.
- A user folder containing the test bench and graphs for the test bench is included to keep these items organized when using the test bench in a larger design project. This folder also makes it easier to "clean up" if you accidentally import the wrong test bench into your project, because you can just delete each of the items under this folder. (Do not delete the folder first, as this only removes the organizing folder, and not the items under it.)
- No global definitions or variables are used. This helps when importing the test bench items into your design project.
- Graphs are named with the industry standard names for the measurements. There may be conflicts with graph names in your design project, however, the Import Project feature can resolve any name conflicts.

D.1. Importing Test Benches

Test benches are intended for import into another project. How you access them depends on your internet access.

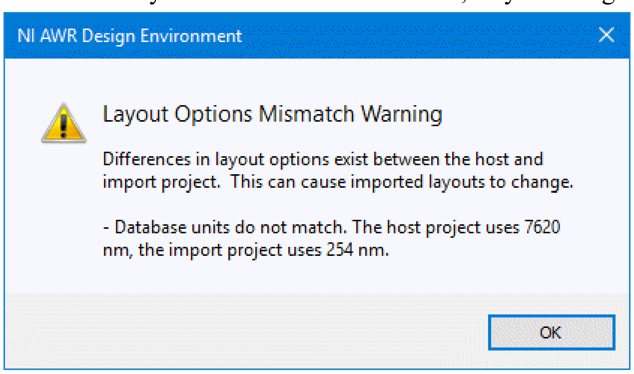
D.1.1. Test Bench Project With Internet Access

When you choose **File > Open Example** and then identify and try to open a test bench project, the corresponding information in the Cadence AWR Knowledge Base opens. This information includes images of the test bench schematic and the display schematic with all of the graphs, and descriptions of the measurements in the graphs and any additional items in the project. When you find a test bench project(s) you want to use, click the **Import Testbench** button to import the necessary items into the currently open project. After import, open the test bench schematic, and set the SUBCKT block's NET parameter to the schematic you want to analyze. Follow the notes in the schematic on what settings to change so the analysis is appropriate for your design.

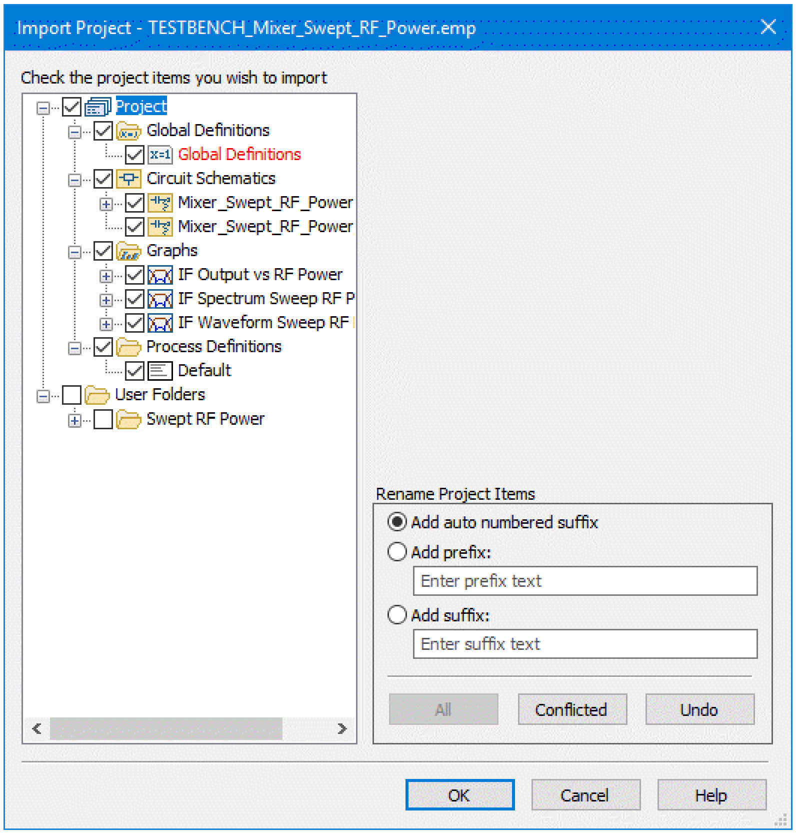
D.1.2. Test Bench Import Without Internet Access

Test bench projects are designed for use with the Import Project feature (see "Importing a Project " for details). If you do not have access to the internet, choose **File > Import Project**, navigate to the installation folder and then the *Examples* folder under it, and find and import the desired TESTBENCH_ project. The following tips smooth the project import process.

When you first import a project (choose **File > Import Project**), a warning might display about layout option differences. There is no layout work in the test benches, so you can ignore this message.

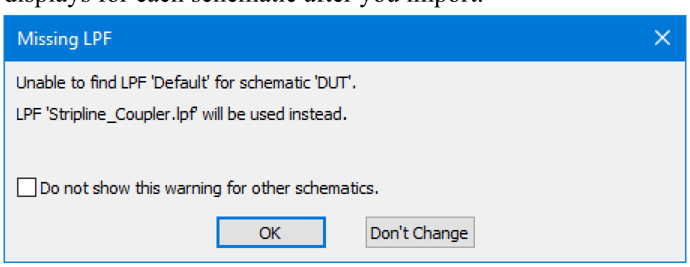


After you click **OK**, an Import Project dialog box displays similar to the following figure.



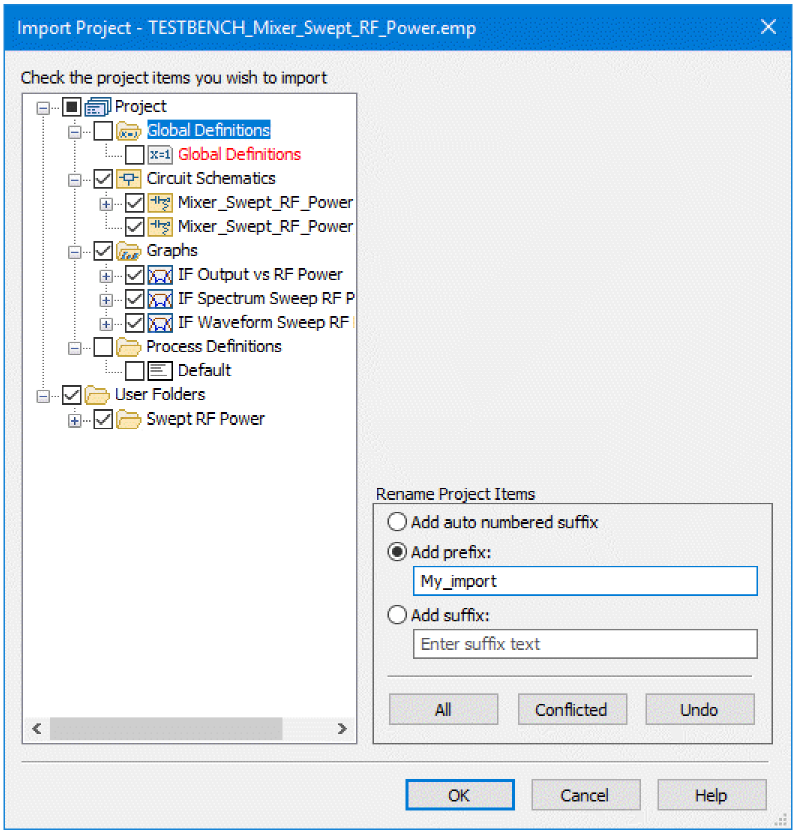
You need to make the following changes:

1. Clear the **Global Definitions** node check box. It displays as conflicted (shown in red), but there is nothing in the **Global Definitions** in the test bench project. Clearing this check box causes the imported test bench to use the original global definitions in your design project.
2. Clear the **Process Definitions** node check box whether or not it displays as conflicted. The test benches should use the process definition in your design project, not that of the test bench project. Note that a message similar to the following displays for each schematic after you import.

   You can ignore these messages; they are just reminders that the new schematics use the process definitions (LPFs) from the design project.
3. Select the **User Folders** node check box so a new folder is created to contain the imported schematics and graphs.
4. Any schematics or graphs that conflict with the current project display in red (in this example all are red). You need to decide how to resolve these conflicts. Cadence recommends that you use the **Add prefix** option with a unique prefix, and then click the **All** button so all items associated with this import have the same prefix, making them easy to identify.

After making these changes, the previous dialog box displays as follows.



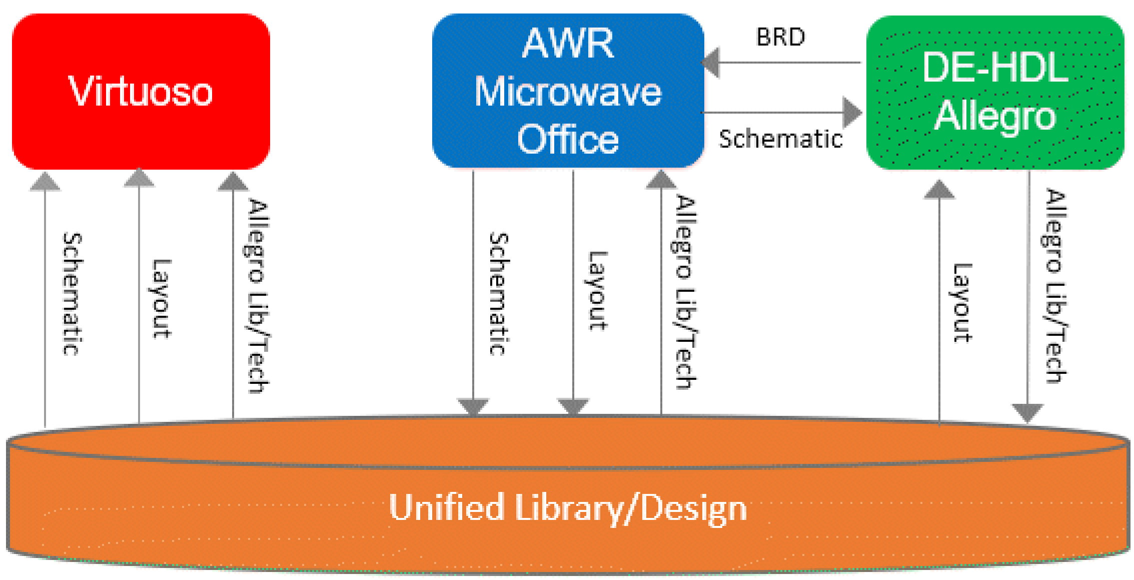
Click **OK** to import the project.

Appendix E. AWR Design Environment Interoperability with Virtuoso and Allegro

Interoperability between Cadence Microwave Office® software, Cadence Virtuoso® software, and Cadence Allegro® Design Entry HDL (DE-HDL) software is a limited release feature. Please contact your sales representative if you have interest in this feature.

Microwave Office software can now transfer technology and design information with Virtuoso and Allegro/DE-HDL software through a Cadence Unified Library. There are various supported flows in the initial release of this interoperability. This appendix describes the capability, intent, use model, and restrictions of this interoperability in its current form. For a list of current restrictions, see "Cadence Unified Library Flow Restrictions/Recommendations".

NOTE: This appendix is limited to the links between Microwave Office, Virtuoso, and Allegro/DE-HDL software, where Microwave Office software is the RF creation platform and the design is sent to Virtuoso or Allegro software for incorporation into a larger system. Other flows exist where the Unified Library allows transfer of data between Virtuoso/Allegro/SiP software directly. These flows are not presented here. The long-term vision of the interoperability between these frameworks is to transfer technology, parts, and design seamlessly, ensuring that the most efficient tool is used for any particular step in the design cycle.



AWR/DE-HDL/Allegro Link

The Microwave Office/DE-HDL/Allegro software link allows designers to create RF IP in Microwave Office software using parts from existing corporate libraries, as well as ensuring the design is done on a manufacturable technology. Starting a design in Microwave Office software with a Cadence Unified Library ensures that:

1. The designer has access to purchasable parts that are seamlessly transferred to a tape-out platform.
2. The STACKUP and layers on which the design is implemented are correct.
3. Manufacturable vias are used in the design process and seamlessly transferred to a tape-out platform.

For details, see “AWR Design Environment/Allegro Interoperability”.

AWR/Virtuoso Link

The AWR/Virtuoso software link allows designers to create IC RF IP in Microwave Office and send the design to Virtuoso software for placement in a larger system for verification and implementation. The IP is transferred through the Unified Library which can be read natively in Virtuoso.

For details, see “AWR Design Environment/Virtuoso Interoperability”.

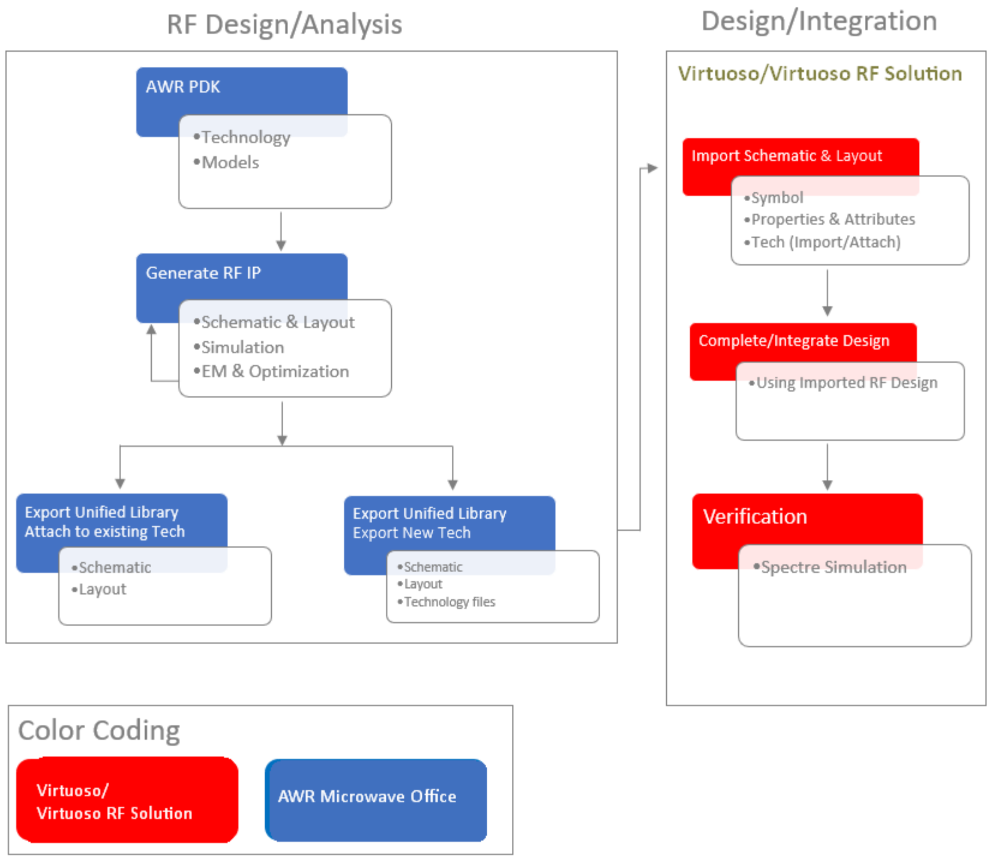
E.1. AWR Design Environment/Virtuoso Interoperability

Interoperability between Microwave Office software and Virtuoso software is a limited release feature. Please contact your sales representative if you have interest in this feature.

E.1.1. Intended Use Model and Benefits

Integrating RF/microwave designs with analog and digital designs can be challenging. These multi-technology and densely populated designs are prone to coupling/crosstalk and other parasitic behavior that can impair overall system performance. To successfully integrate RF/microwave design data with mixed-signal designs, design software needs to exchange design data efficiently. AWR Design Environment software offers an RF/microwave design creation platform with export functionality to provide a pathway to the Virtuoso IC design tool framework.

Microwave Office software provides the platform for RF/microwave design entry, circuit/system/EM analysis, and optimization. The result is an electrical design with the layout and IC stackup information necessary to ensure accurate prediction of manufactured device performance. Transferring the electrical design, physical layout and stackup information from AWR software to Virtuoso software eliminates the need for manual design re-entry; reducing time, costs, and the potential for errors. The foundation of AWR software and Virtuoso software interoperability is the Unified Library. Unified Libraries are common databases that both AWR and Virtuoso software can read, and contain the technology, device, and design information required to properly analyze, verify, and manufacture the design. The following diagram shows the export flow for RF/microwave designs.



For a list of current interoperability restrictions, see "Restrictions with Virtuoso Interoperability ".

E.1.2. Microwave Office Software to Virtuoso Software Flow

The Microwave Office software to Virtuoso software interoperability flow includes several steps to exchange design data between programs. This appendix provides an overview of the process, but focuses on the steps within Microwave Office software. For more information about the steps within Allegro PCB design applications, see the associated documentation.

The Microwave Office software to Virtuoso software flow includes the following steps:

1. Create a design in Microwave Office software.
2. Export the AWR design to the Cadence Unified Library with the **Export New Tech** or **Attach to Existing Tech** option.
3. Open the exported Cadence Unified Library in Virtuoso/Virtuoso RF Solution software.
4. Integrate and complete the design in Virtuoso software.
5. Verify the exported design with the Cadence Spectre® Simulation Platform.

Design Flow Benefits

The Microwave Office software to Virtuoso software flow provides the following benefits over a traditional RFIC/microwave design flow:

- Microwave Office schematic and layout designs are easily exported and opened in Virtuoso software.
- AWR exported design IP includes Virtuoso tech files for easy integration of exported schematics and layouts into Virtuoso and Virtuoso RF Solution software.
- Exported Microwave Office schematic elements can be simulated using the Spectre Simulation Platform in Virtuoso software.
- Microwave Office designs containing S-parameter files are fully supported and do not require extra steps.

E.1.3. Exporting the Microwave Office Design to a Unified Library

After creating a design in Microwave Office software, you can export the RF schematic and layout data to a Unified Library for use in Virtuoso/Virtuoso RF Solution design applications, where an RF designer can integrate the RF IP along with the analog and digital designs.

For a list of current export restrictions, see "Restrictions with Virtuoso Interoperability ".

To export a Cadence Unified Library design:

1. In the Project Browser, right-click the top-level schematic you want to export and choose **Cadence Unified Library Export**. The Cadence Unified Library Export dialog box displays.
2. Export a design from AWR software to Virtuoso/Virtuoso RF Solution software using one of the following **Tech Option** choices:
   - Export New Tech
   - Attach to Existing Tech

E.1.3.1. Exporting Designs from AWR to Virtuoso/Virtuoso RF Solution Using Export New Tech Option

If the PDK used for the AWR design is not available in Virtuoso software, select **Export New Tech** as the **Tech Option**. The design library that is generated includes all of the necessary technology files for integrating the exported design into Virtuoso/Virtuoso RF Solution software. To select this option:

1. Click the **Presets** button and choose **IC-Virtuoso** to set options to create a tech file with IC fabric.
2. Specify the **Export Directory**. Spaces are not supported in the file path.
3. Specify the **Design Library Name** (the default is the exported schematic name) and **Design Cell Name**.
4. Ensure that **Tech Option** is set to **Export New Tech**, then click **OK** to start the export and close the dialog box.

The following technology files are generated when using the **Export New Tech** option:

- *tech.db* - Virtuoso tech file. AWR Process Technology information in Virtuoso format (fabric type, layer definitions, constraints, layer connectivity)
- *display.drf* - Virtuoso layout display resource file containing layer color and layer pattern information
- *cds.lib* - Virtuoso library definition file
- *MWOspectreNetlist.il* - Spectre Simulation Platform netlist procedures for exported Microwave Office elements
- *libInit.il* - Virtuoso library initialization script. File is executed the first time the exported library is loaded. Loads *display.drf* and *MWOspectreNetlist.il* files

NOTE: The AWR PDK may require a one time step of compiling Windows model files on Linux to make them available for Spectre Platform simulation in Virtuoso software

E.1.3.2. Exporting Designs from AWR to Virtuoso/Virtuoso RF Solution Using Attach to Existing Tech for Virtuoso PDK Re-Use

If the PDK used for the AWR design has a common PDK available in Virtuoso software, select **Attach To Existing Tech** as the **Tech Option**. To select this option:

1. Click the **Presets** button and choose **IC - Virtuoso** to set options to create a tech file with IC fabric.
2. Specify the **Export Directory**. Spaces are not supported in the file path.
3. Specify the **Design Library Name** (the default is the exported schematic name) and **Design Cell Name**.
4. Ensure that **Tech Option** is set to **Attach To Existing Tech**.
5. Specify in **Attach To Technology** the technology library name in the Unified Library. This setting is pre-populated with the **Technology Library** specified during the Cadence Unified Library Import process. Click **OK** to start the export and close the dialog box.

E.2. AWR Design Environment/Allegro Interoperability

Interoperability between Microwave Office software and Allegro/DE-HDL software is a limited release feature. Please contact your sales representative if you have interest in this feature.

E.2.1. Intended Use Model and Benefits

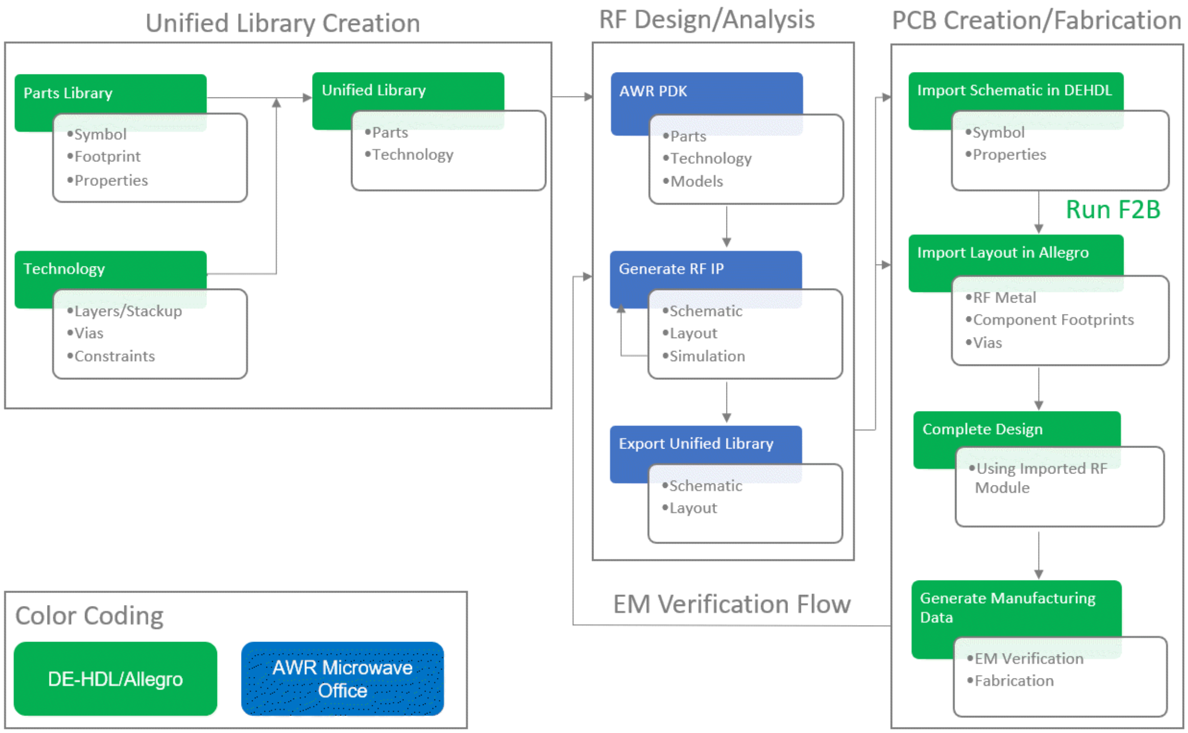
Incorporating RF/microwave, analog, and digital design elements together on the same PCB is challenging. These multi-layer PCBs are densely populated with high-speed data lines and RF circuitry and are prone to coupling/crosstalk and other parasitic behavior that can impair system performance. To successfully integrate RF/microwave content and mixed-signal designs, PCB layout tools and RF circuit design software must exchange design data efficiently. The AWR Design Environment platform offers an RF/microwave design creation environment with import and export functionality to provide a pathway to and from Cadence Allegro® PCB design tools.

Microwave Office software supports a broader PCB subsystem design flow. The integrated flow between Microwave Office software and Allegro PCB design applications operates as a standalone solution or part of an enterprise-level PCB system. Microwave Office software provides the platform for RF/microwave design entry, circuit/system/EM analysis, and optimization. The result is an electrical design with the layout and PCB stackup information necessary to ensure accurate prediction of device performance. Transferring this layout and stackup information into the Allegro PCB layout and routing platform eliminates the need for manual design reentry, thus saving time, costs, and the potential for errors. Microwave Office software compatibility with Allegro PCB design applications ensures accurate and fast simulation of complex PCB designs.

The foundation of Microwave Office and Allegro software interoperability is the Unified Library. Unified Libraries are common databases that Microwave Office and Allegro/DE-HDL software can read. They contain the technology, component, and design information required to properly analyze, verify, and manufacture a design. The Unified Library is generated from Allegro/DE-HDL software.

NOTE: To ensure symbol compatibility, you must use the Symbol Compatibility guidelines when generating the DE-HDL symbols.

The following figure shows a design flow for RF/microwave PCBs.



For a list of current interoperability restrictions, see "Restrictions with Allegro Interoperability".

E.2.2. Microwave Office Software to Allegro PCB Software Flow

The Microwave Office software to Allegro PCB software interoperability flow includes several steps to create shareable IP in Microwave Office software with proper library parts and technology for use in Allegro PCB design software. This appendix provides an overview of the process, but focuses on the steps within Microwave Office software. For more information about the steps within Allegro PCB design applications, see the associated documentation.

The Microwave Office software to Allegro PCB software flow includes the following steps:

1. In Allegro/DE-HDL software, translate the Allegro library of discrete components to Cadence Unified Library format for use by Microwave Office software.
2. Create the AWR PDK from the Cadence Unified Library.
3. Create a design in Microwave Office software using the technology and parts from the AWR PDK.
4. Export the Cadence Unified Library from the Microwave Office software design for use by Allegro PCB design applications.
5. Import the Microwave Office design (IFF) into Allegro/DE-HDL software. The import process creates an identical schematic in Allegro/DE-HDL software, which when packaged, generates a netlist for import into Allegro PCB Editor for layout creation.
6. Import the Microwave Office layout (.OA) into Allegro PCB Editor.
7. Run design synchronization in the Allegro Design Authoring application to verify connectivity between RF schematic and layout and save the RF layout as a module you can use in other PCB designs.

Design Flow Benefits

The Microwave Office software to Allegro PCB software flow provides the following benefits over a traditional RF/microwave PCB design flow:

- Allows design in the context of Allegro/DE-HDL parts and technologies in Microwave Office software.
  - Parts used in Microwave Office software are Allegro/DE-HDL parts that have identical symbols, footprints, and properties. They can be purchased and do not need to be replaced once in the manufacturing framework.
  - The technology information used is identical to the manufacturing technology, and the STACKUP information is provided for EM simulations.
  - Manufacturable vias are available for simulation and placement, and are transferred with the design, so no replacement is required.
  - Physical design constraints are available within Microwave Office for dynamic voiding of ground/power planes.
- Provides a seamless data transfer from Microwave Office software to Allegro/DE-HDL of all library parts, RF metal, and vias, and eliminates time-consuming and error-prone manual re-entry or replacement of parts/vias.
- Allows EM verification of the performance of the RF design after it is incorporated with the rest of the PCB.

E.2.2.1. Creating an AWR PDK from a Unified Library

This step requires that the Unified parts and technology library from Allegro/DE-HDL software is already created. See the Allegro/DE-HDL documentation for information on generating a Unified Library if needed.

To utilize parts common to both AWR software and Allegro PCB design applications, you import the Cadence Unified Library from Allegro software using the Cadence Unified Library Import wizard to wrap the Unified Library into a AWR PDK.

To import a Cadence Unified Library and create an AWR PDK:

1. In the Project Browser **Wizards** node, double-click the **Cadence Unified Library Import** Wizard.

   The Cadence Unified Library Import dialog box displays.
2. Specify the **Import Type** as **Allegro Library**.
3. Browse to the **cds.lib File** that contains the library/libraries you want to add to the PDK. You can edit this text file in a text editor. Spaces are not supported in the file paths.
4. From the drop-down list, select the **Technology Library** to import. This library should contain the *tech.db* which has the layer and STACKUP information, and the via definitions to use in the design.
5. From the drop-down list, select the **Display Name** to specify the display selection in the *.drf* file in the specified **Technology Library** and control the colors/fills in the Microwave Office *.lpf* file.
6. Specify the **Output Directory** for the AWR PDK. Spaces are not supported in the file paths.
7. Specify the **PDK Name** and **PDK Version**, and then click **OK**. For more information about these options, see "Cadence Unified Library Import Wizard Dialog Box: Options Tab".

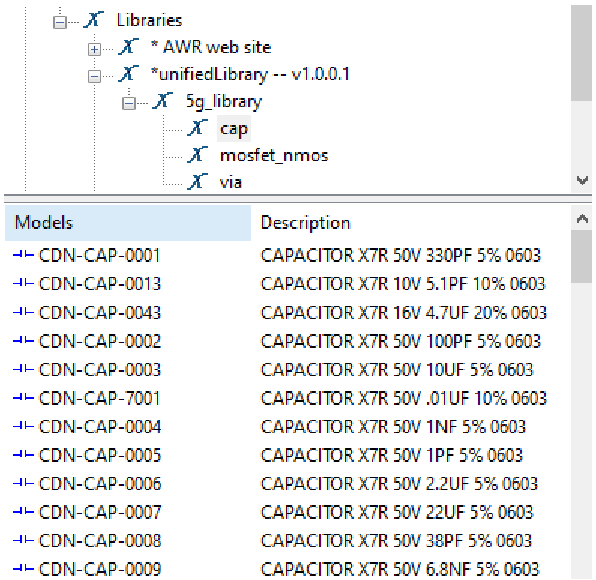
   A message displays when the process is complete.

For more information about the Cadence Unified Library Import dialog box options, see "Cadence Unified Library Import Wizard Dialog Box: Options Tab".

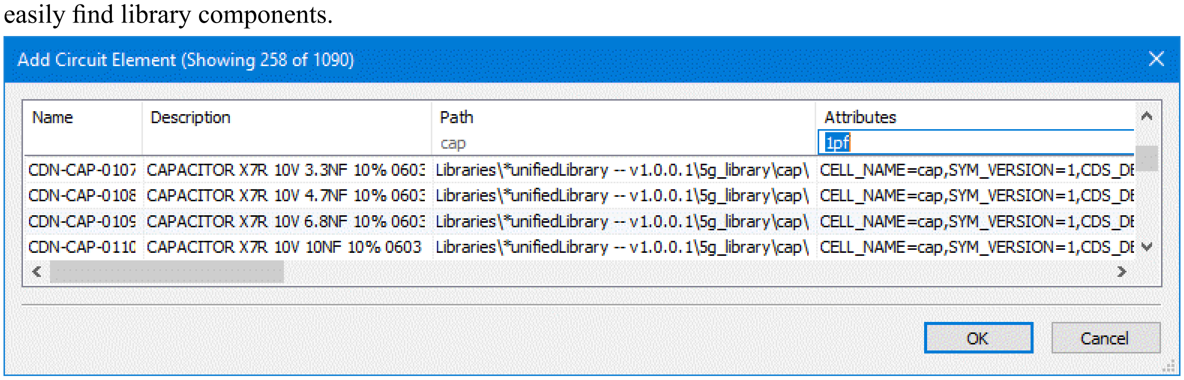
E.2.2.2. Designing with a Unified Library

After creating an AWR PDK from the Cadence Unified Library, you can use the technology (layers, STACKUP/material information, vias, physical constraints) and parts (schematic symbols, properties, footprints, models[1]) to create a schematic with a corresponding layout, and then simulate the design. The process is very similar to traditional RF PCB design, except that all parts/components used in the design are from the library available in the Elements Browser **Libraries** node. For more information on designing with a Unified Library PDK, see Model Association.

In the Elements Browser (click the **Elements** tab), you can view the elements in the PDK by opening the **Libraries** node and then selecting the nodes beneath the **PDK Name**. All parts/components you use in your design must come from this library.



To select components from this library, you can drag them to the schematic or choose **Draw > More Elements** to display the Add Circuit Element dialog box, where you can filter/sort on the **Name**, **Description**, **Path**, or **Attributes** columns to easily find library components.



Model Association

Models can be defined in multiple ways, and many model types are supported in the flow. In some situations, an ideal model can be automatically created for capacitors, inductors, and resistors. Automatic model creation requires a

[1]Allegro/DE-HDL part libraries often do not have model views, so you may need to associate some parts to models. The designer/librarian can manage this mapping at the Unified Library level or at the Microwave Office PDK level.

PART_NAME column in the *part.csv* file that has recognizable part names. For automatic model creation, the model definition is written into a *model_ex.csv* file alongside the *part.csv* file. For all other situations, you or the PDK librarian need to explicitly add the desired modeling information to *model_ex.csv*. If additional model types are desired, you can append the *model_ex.csv* file with this information and the models will be available for use.

Models that are supported are S-parameters, and any Microwave Office model (including any built-in model in Microwave Office or PDK-defined models). Models defined in *part.csv*/*part_ex.csv* files are tunable/optimizable through the rows defined in *part.csv*, or through the variants of a specific value, and all parameters, properties, and layouts are updated as the part changes. Parameter tolerance information can also be read from the *part.csv* file for yield analysis.

Supported Model Flows

For increased flexibility, you can define models in multiple ways/places depending on the flow you want to adopt. The supported flows are the Corporate Library Model Flow or the Ad-hoc Model Flow.

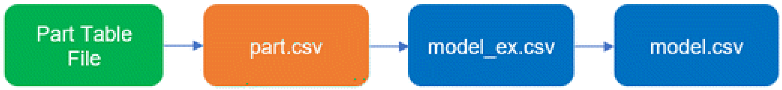
<u>Corporate Library Model Flow:</u> In this flow, model information is added directly to the PCB library as properties in the part table file. When the PCB library is augmented with the Unified Library Views of the components, the model information propagates and is available for simulation in Microwave Office software. The different files involved in the modeling of the parts are:

- The *part.csv* file is the primary definition of the part, which requires certain columns to adequately describe the part. It often does not include any modeling information. The columns that are required and/or recommended are:
  - CDS_DEVICE_TYPE - a unique identifier used to identify a row in the *part.csv* file. Each row represents a unique part. This column is required.
  - PART_NAME - an identifier used to describe the part. As described previously, certain part names are recognized automatically and used to automatically generate an ideal model.
  - VALUE - describes the value of the part (such as the capacitance value). For recognized PART_NAME components, the VALUE generates ideal models (for capacitors, resistors and inductors only).
- The *model_ex.csv* file stores modeling information you add.
- The *model.csv* file is a reformatted version of the *model_ex.csv* file that is actually consumed by the simulation tools. Note that *model_ex.csv* is stored in an Excel friendly/compatible format, which allows you to easily add model information to it using Excel. The *model.csv* file is automatically generated from the *model.csv* used by Microwave Office and other Cadence tools.

Microwave Office software always uses the information in the *model.csv* file for the model used in the Microwave Office simulations, but it is more convenient to maintain the *model_ex.csv* file and have the system automatically generate *model.csv*. The actual modeling data is defined with the following columns in either *part.csv* or *model_ex.csv* (which is translated to *model.csv*).

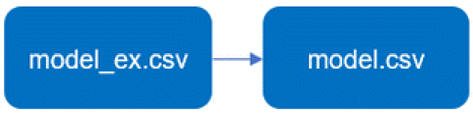
- AWR_MODEL
- AWR_MODEL2
- SPARAM_FILE

The model columns can be in either in the *part.csv*, or the *model_ex.csv* file. If the model columns are in *part.csv*, a *model_ex.csv* file is automatically created with the same model column as the *part.csv* file. This allows you to add to the information in *model_ex.csv* without changing the information in *part.csv* (which is important in some library maintenance flows).



The benefit of the Corporate Library Model Flow is that the model information is added and stored at the source of the library. If the library is changed, any updates are pushed to end users.

Ad-hoc Model Flow: In the Ad-Hoc Model Flow, model information is added by the end-user (typically the RF/Microwave engineer) directly to the PDK that is created when the Unified Library is translated to Microwave Office. The information is added to the *model_ex.csv* file for each cell in the Unified Library and is then available for simulation in Microwave Office.



The Ad-Hoc Model Flow is for users that do not wish to store RF model information in their PCB libraries or generally rely on the RF/Microwave engineer to select the appropriate models. The Ad-Hoc Model Flow does have some downsides when compared to the Corporate Library Model Flow. If the library is updated, it may require re-associating models with the available parts. While this flow does have some flexibility and may seem more efficient, it is important to consider the cost of library maintenance.

Adding Models

This section provides information about adding models to the *model_ex.csv* file with the appropriate syntax. To add models to the part table file, the same syntax applies, but the format of the part table file must be followed. See the DE-HDL documentation for that information. The following figure shows the different parts of the *model_ex.csv* file.

A	B	C	D	E	F	G	H
CDS_DEVICE_TYPE	PART_NAME	VALUE	VOLTAGE	TOLERANCE	AWR_MODEL	AWR_MODEL2	SPARAM_FILE
Device Type	PART_NAME	VALUE	VOLTAGE	TOLERANCE	Ideal Cap	AWR Fancy Model	s-parameter
CAP_0603-CDN-CAP-0001__330PF,50V,5%,0603	CAP	330PF	50V	5%	CAP C~330p	CHIPCAP C=330p	330_pF_sparameter.s2p
CAP_1005N-CDN-CAP-0013__5__1PF,10V,10%,1005N	CAP	5.1PF	10V	10%	CAP C~5.1p	CHIPCAP C=5.1p	5p1_pF_sparameter.s2p
CAP_1608N-CDN-CAP-0005__1PF,10V,40%,1608N	CAP	1PF	10V	40%	CAP C~1p	CHIPCAP C=1p	1_pF_sparameter.s2p
CAP_0603-CDN-CAP-0006__2__2UF,10V,40%,0603	CAP	2.2UF	10V	40%	CAP C~2.2e+06p	CHIPCAP C=2.2u	2p2_uF_sparameter.s2p

Each *model_ex.csv* file is unique, depending on the source PCB part table data. Each file has, at a minimum, a CDS_DEVICE_TYPE column which makes that part unique.

Microwave Office Model

Microwave Office models, including PDK models, are defined with the following:

- 1st row is "AWR_MODEL" column header (you can also use AWR_MODEL2 if you want to support a second AWR model that can be selected from the part instance).
- 2nd row is a description of your choosing. Note that this displays in the model selector of the part and allows you to choose which simulation model to use since multiple model types are supported for a single component.
- 3rd - nth is the per instance model information (each row defines a unique part along with the model used for that part).
  - Option to support setting model parameter values
  - Option to support pin mapping

A simple example for an ideal capacitor is:

```
CAP C=100p
```

NOTE: If you want to add multiple models per part, you need multiple columns. This is supported by appending a numerical index to the 1st row column header (for example: AWR_MODEL, AWR_MODEL2). Currently, a maximum of two AWR models per part are supported.

S-Parameter Model

- 1st row is "SPARAM_FILE" column header.
- 2nd row is a description of your choosing. Note that this displays in the model selector of the part and allows you to choose which simulation model to use since multiple model types are supported for a single component.
- 3rd - nth is the S-parameter file name.
  - All S-parameters exist in a folder in the same cell called "sparameters", so no path is required in the *.csv* file.
  - These can be stored with the original library if you are using the Corporate Library Model Flow or in the PDK if using the Ad-Hoc Model Flow.
  - Option to support pin mapping

Additional SHELL_CSV Functionality

Automatic Ideal Model Creation

The first time a model is placed from a particular cell, the *part.csv* file is evaluated, and if the appropriate information exists, and no other model information is present, ideal models for each part are added. These ideal models are added to the *model_ex.csv* file. This capability only functions for simple R,L,C components and requires that the PART_NAME column start with either RES, CAP, or IND (case insensitive) as well as having a VALUE column with an appropriate value for the component type.

If TOLERANCE information exists as well, the component is set up for yield analysis in Microwave Office software. Note in the previous figure that the "Ideal Cap" is a CAP model with a C value specified. Instead of an "="" for the parameter value, a "~" is used instead, which specifies that this parameter is set up for yield analysis based on the tolerance specified in the TOLERANCE column.

Parameter Mapping

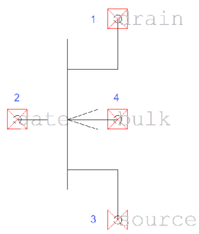
If the AWR Models need to have parameters set, you can do so in the model specification by using <parameter name>=<parameter value> after the model name. Multiple parameters are space-separated and unit multipliers are supported. The supported multipliers are `f, p, n, u, m, k, M,` and `G` (so you can use `C=1p` for `C=1e-12`).

Pin Remapping

If the simulation model and symbol graphics pin mapping is not 1-to-1, you can remap them in the model specification by using the positional ordering for the model with a map to the symbol pin number in parenthesis before the model specification in the model column.

CDS_DEVICE_TYPE	PART_NAME	AWR_MODEL
Device Type	PART_NAME	AWR_MESFET
POWER_NMOS-NMOS65W	POWER_NMOS	(1 3 2 4) GENERIC_LDMOS

In this example the symbol has the following pin order (D G S B).



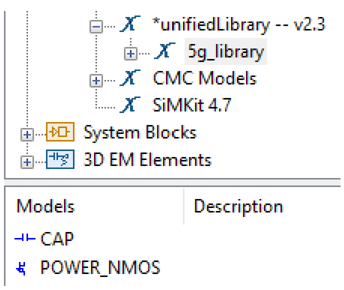
The underlying model has a pin order of (D S G B). To ensure proper simulation results, the Gate and Source must be swapped by remapping the symbol pin number to the appropriate positional order of the underlying model.

Purpose	Symbol Pin Number	Model Pin Position
Drain	1	1
Gate	2	3
Source	3	2
Bulk	4	4

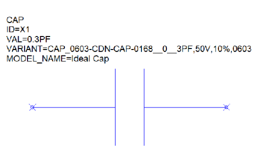
Entering (1 3 2 4) remaps the Gate (symbol pin 3) and Source (symbol pin 2) to a positional order of 2 and 3 respectively, which is how the model is coded.

Using a SHELL_CSV-Based Model in Design

Placing Parts



Model/Part Selection and Simulation



VARIANT only exists if the same VALUE component exists multiple times in the *part.csv* file. Typically different VARIANTS might have difference voltage ratings, tolerance, or sizes, for example.

The combination of VALUE and VARIANT selects the row and ensures the correct model and footprint is used in the design based on the your selections.

MODEL_NAME controls which model representation is used in simulation. For example, you might have ideal models and an S-parameter representation of the part; this parameter selects which one to use.

Working with Library Elements

The components in the Unified Library are slightly different than standard Microwave Office elements. They are technology independent layout pCells (TILPs) with enhanced functionality. Some of this functionality is the same as built-in functionality in Microwave Office, such as the rotation specification, and some should not be changed. Changing the other parameters results in missing and/or incorrect footprints.

To access the Unified Library layout cell parameters, select the element in the layout, then right-click and choose **Shape Properties**. In the Cell Options dialog box, click the **Parameters** tab to display the list of available parameters. From this list of parameters, you should only alter "mirrored", "offset", or "rotation".

- Mirrored
  - 0 = On Top of the board
  - 1 = On Bottom of the board
- Offset
- Rotation
  - Value of the rotation in degrees or radians. It can be coupled with the Microwave Office rotation and the two combined to a correct representation in the Unified Library Export.

EM Analysis

If EM analysis needs to be performed, it can be done through EM extraction. The STACKUP is available for use in the PDK Global Definitions document. See “*EM: Creating EM Structures with Extraction*” for information on how to set up an EM extraction.

E.2.2.3. Exporting the Design to a Unified Library for Use in Allegro/DE-HDL

After creating a design in Microwave Office software, you can export the RF schematic and layout data to a Unified Library for use in Allegro PCB design applications, where a PCB designer can assemble the rest of the board.

For a list of current export restrictions, see “Restrictions with Allegro Interoperability”.

To export the design data as a Cadence Unified Library:

1. In the Project Browser, right-click the top-level schematic you want to export and choose **Cadence Unified Library Export**. The Cadence Unified Library Export dialog box displays. For more information about this dialog box, see “Export Cadence Unified Library Dialog Box”.
2. Click the **Presets** button and choose **Board - Allegro** to set options that are required for the AWR/Allegro/DE-HDL link.
3. Specify the **Export Directory**. Spaces are not supported in the file path.
4. Specify the **Design Library Name** and **Design Cell Name**.
5. Ensure that **Tech Option** is set to **Attach To Existing Tech**. This setting should auto-set when exporting from a project that uses a Unified Library PDK.
6. Ensure that **Attach To Technology** specifies the technology library name in the Unified Library. This setting is pre-populated with the value specified as the Technology Library during the Cadence Unified Library Import process.
7. Ensure that **Polygon Cutouts** is set to **Holes**.
8. Ensure that **Polygon Arcs** is set to **Native**, then click **OK**.

A message displays when the export process is complete, and a Unified Library representation of the design is created in the **Export Directory** path. You can now import this data into Allegro/DE-HDL. See the Allegro/DE-HDL documentation for information on importing design data if needed.

E.2.2.4. EM Verification Flow

RF designs are typically designed independently of the rest of the PCB, so you should ensure that the RF performance remains intact when assembled with the other parts of the PCB. To do so, the completed geometry and material information is required, and you should perform an EM analysis on the metal/dielectrics, then verify the performance of the RF circuit with the EM results and the original component list. Microwave Office software can import the manufacturing data from Allegro software to perform this type of verification, and provides tools to assist in preparing the geometry for EM analysis. For more information about this process, see "PCB Import Wizard".

EM extraction can be performed using the STACKUP in the PDK's Global Definitions document available in the Project Browser (click the **Project** tab) under the **Global Definitions** node. See "*EM: Creating EM Structures with Extraction*" for more information.

E.3. Interoperability Restrictions

The following restrictions apply to AWR Design Environment software interoperability with Virtuoso or Allegro software, or both.

E.3.1. Cadence Unified Library Flow Restrictions/Recommendations

- Flow Considerations
  - Data exported from Microwave Office software is read-only in Virtuoso and Allegro software.
  - Connectivity or parts changes must be made in the AWR Design Environment platform
- Namespace rules
  - Spaces are not supported in Cadence Unified Library paths, AWR PDKs created via the Cadence Unified Library Import Wizard, or designs exported from the AWR Design Environment using the Cadence Unified Library Export.
  - Colons (":") are not supported in model names
- Unsupported Simulation/Layout Features
  - Switch Views
  - Layout and Shape Modifiers
  - 3D shapes (for example, housings or SMA connectors)
  - S-parameters extracted from EM documents are not transmitted with the design
  - AWR graphs and measurements do not export

E.3.2. Restrictions with Virtuoso Interoperability

Required Software Versions

1. AWR = V16.0 or later
2. Virtuoso = ICADVM20.1 ISR17 and IC6.1.8 ISR17 or later
3. Spectre Simulation Platform = 20.10 ISR7

Virtuoso PDK Re-Use Requirements

- AWR = V16.03 or later
- Updating parameter values in the APLAC_PROC element in AWR always requires running a new simulation.

Flow and Best Practices Considerations

- Data exported from AWR software is read-only in Virtuoso software.
- Flow use is to export only design IP, not test benches containing sources.
- All exported designs require a Backside Metal or Backside Via layer.
- Design and parameter changes must be made in AWR software, then re-exported to Virtuoso software.
- Microwave Office PORT elements are exported as I/O pins in Virtuoso software.
- Avoid connecting PORT elements directly to element nodes.
- Microwave Office exported design data is not fully XL compliant in Virtuoso software.
- Document names should be lower case letters.
- AWR schematic names must be unique for export.

Restrictions

- You cannot load more than one Virtuoso Design Kit (VPDK) within the same AWR Design Environment project.
- You cannot use multiple versions of the same Cadence Unified Library or VPDK within the same AWR Design Environment project.
- The VPDK installation, AWR Design Environment installation, and AWR project file must be in locations that do not contain spaces in their respective paths.

Unsupported Topology

- Some AWR PDK elements with implicit grounds.
- UPPER CASE letters. Upper letters case are changed to lower case on export.
- AWR Switch Views.
- Some Layout and Shape Modifiers.
- MDIF data files.
- AWR EM structures in Microwave Office schematics.
- Package fabric (Preset "Package - Virtuoso") designs must be flat (no hierarchy) with all elements instantiated at the top level.
- Package fabric (Preset "Package - Virtuoso") designs may only contain TILPs connected by traces.

Unsupported Elements

- X-models with custom stackups (Shipping X-model stackups **are** supported.).
- PORT_TN
- NEG
- NPORT_MDIF
- PORT_NAME (when using this element in a subcircuit, you cannot connect by named parameter.)
- Nonlinear measurement ports such as Port1/Port2

E.3.3. Restrictions with Allegro Interoperability

Required Software Versions

1. AWR = V16.0 or later
2. Allegro/DE-HDL = 17.4-2019 QIR3 or later

Flow

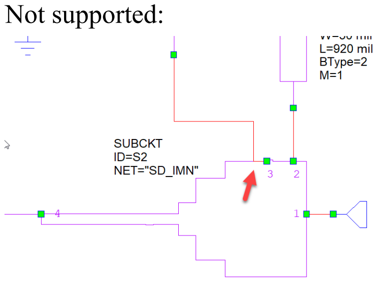
- Data exported from AWR software is read-only in Allegro/DE-HDL software.
- Do not export simulation test benches; maintain a distinction between implementation and simulation schematics.
- Connectivity, design, or parts changes must be made in the AWR Design Environment platform.
- Technology or library component changes must be made in Allegro, and the AWR PDK regenerated.
- Unit changes are not supported within Microwave Office software.
- Schematics are interoperable with DE-HDL software only.
- Only designs created with a Unified Library are supported.
- Schematic names should be lower case letters.
- Schematic names must be unique.

Unsupported Elements:

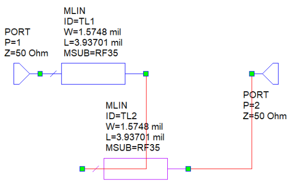
- PORT_NAME
- NCONN
- Non-manufacturable parts (for example bias supplies or ideal lumped elements) other than PORTx and GND elements
- Elements with implicit ground nodes (for example VIA1P)

Unsupported Topology:

- Switch Views
- Non-planar EM structures (for example, connectors and waveguides)
- Wires cannot overlap a symbol boundary (this is different from crossing a symbol boundary).
  - Not supported:



  - Supported:



- Allegro split symbols, sizeable parts, and asymmetric parts.

Other:

- AWR system drawing layers are not sent to Allegro
- Special characters (such as "&" or "$") are not permitted in Element IDs
- Nodes/pins on logical symbols coming from Allegro/DE-HDL software must be on a grid of 100.

Index

Symbols

A

B

C

F

G

H

I

K

L

M

W

X

Y

Z