cādence®

Virtuoso Analog Design Environment XL User Guide

Product Version 6.1.6
August 2014

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

Contents

16 Working with the Simulation Setup

Preface

The Virtuoso Analog Design Environment XL (ADE XL) is an advanced design and simulation environment.

This manual describes how you can set up tests, simulate your designs, and analyze output in the ADE XL environment. The information presented in this manual is intended for integrated circuit designers and assumes that you are familiar with analog design and simulation.

This preface describes the following:

Licensing in ADE XL

For information on licensing in ADE XL, see *Virtuoso Software Licensing and Configuration Guide*.

Related Documents for ADE XL

The following documents provide more information about the topics discussed in this guide.

Installation, Environment, and Infrastructure

- For information on installing Cadence products, see the *Cadence Installation Guide*.
- For information on the Virtuoso design environment, see the *Virtuoso Design Environment User Guide*.
- For information on database SKILL functions, including data access functions, see the *Virtuoso Design Environment SKILL Reference*.
- For information on library structure, the library definitions file, and name mapping for data shared by multiple Cadence tools, see the *Cadence Application Infrastructure User Guide*.

Technology Information

- For information on how to create and maintain a technology file and display resource file, see the *Virtuoso Technology Data User Guide* and the *Virtuoso Technology Data ASCII Files Reference*.
- For information on how to access the technology file using SKILL functions, see the *Virtuoso Technology Data SKILL Reference*.

Virtuoso Tools

- *Virtuoso Schematic Editor L User Guide* and *Virtuoso Schematic Editor XL User Guide* describe Cadence's schematic editor.
- *Virtuoso Spectre Circuit Simulator and Accelerated Parallel Simulator User Guide* and *Virtuoso Spectre Circuit Simulator Reference* describe Cadence's Spectre analog circuit simulator.

- *Virtuoso Spectre Circuit Simulator and Accelerated Parallel Simulator RF Analysis User Guide* describes Cadence's RF circuit simulation option.
- *Virtuoso UltraSim Simulator User Guide* describes Cadence's multi-purpose single engine, hierarchical simulator, designed for the verification of analog, mixed signal, memory, and digital circuits.
- *Virtuoso AMS Designer Simulator User Guide* describes Cadence's AMS mixed-signal circuit simulator.
- *Virtuoso Parasitic Estimation and Analysis User Guide* describes parasitic simulation of analog signal circuits using the Virtuoso Analog Design Environment and Schematic Editor.
- *Virtuoso Visualization and Analysis Tool User Guide* contains product information for viewing waveforms and post-processing simulation results.
- *Component Description Format User Guide* describes Cadence's Component Description Format (CDF) for describing parameters and the attributes of parameters of individual components and libraries of components.
- *Analog Expression Language Reference* contains concept and reference information about the Analog Expression Language (AEL).
- *Virtuoso Design Environment Migration Guide* describes the release level changes and migration related information.

Additional Learning Resources

Cadence provides various Rapid Adoption Kits that you can use to learn how to employ Virtuoso applications in your design flows. These kits contain workshop databases, designs, and instructions to run the design flow.

Cadence offers the following training course on the Virtuoso Electrically Aware Design flow:

- Virtuoso Analog Design Environment
- Virtuoso Schematic Editor
- Analog Modeling with Verilog-A
- Behavioral Modeling with Verilog-AMS
- Real Modeling with Verilog-AMS
- Spectre Simulations Using Virtuoso ADE

- Virtuoso UltraSim Full-Chip Simulator
- Virtuoso Simulation for Advanced Nodes
- Virtuoso Electrically-Aware Design with Layout Dependent Effects

For further information on the training courses available in your region, visit the Cadence Training portal. You can also write to training_enroll@cadence.com.

Note: The links in this section open in a new browser. The course links initially display the requested training information for North America, but if required, you can navigate to the courses available in other regions.

Third Party Tools

To view any `.swf` multimedia files, you need:

- A SourceLink Login.
- Flash-enabled web browser, for example, Internet Explorer 5.0 or later, Netscape 6.0 or later, or Mozilla Firefox 1.6 or later. Alternatively, you can download Flash Player (version 6.0 or later) directly from the Adobe website.
- Speakers and a sound card installed on your computer for videos with audio.

Typographic and Syntax Conventions

This list describes the syntax conventions used in this manual.

`literal` — Nonitalic words indicate keywords that you must enter literally. These keywords represent command (function, routine) or option names.

`argument (z_argument)`

Words in italics indicate user-defined arguments for which you must substitute a name or a value. (The characters before the underscore (_) in the word indicate the data types that this argument can take. Names are case sensitive. Do not type the underscore (*z_*) before your arguments.)

`[ ]` — Brackets denote optional arguments.

...	Three dots (...) indicate that you can repeat the previous argument. If you use them with brackets, you can specify zero or more arguments. If they are used without brackets, you must specify at least one argument, but you can specify more.
`argument...`	Specify at least one, but more are possible.
`[argument]...`	Specify zero or more.
`,...`	A comma and three dots together indicate that if you specify more than one argument, you must separate those arguments by commas.

If a command line or SKILL expression is too long to fit inside the paragraph margins of this document, the remainder of the expression is put on the next line, indented.

When writing the code, put a backslash (\) at the end of any line that continues on to the next line.

SKILL Syntax Examples

The following examples show typical syntax characters used in SKILL. For more information, see the *Cadence SKILL Language User Guide*.

Example 1

`list( g_arg1 [g_arg2] ...) => l_result`

Example 1 illustrates the following syntax characters.

`list`	Plain type indicates words that you must enter literally.
g_arg1	Words in italics indicate arguments for which you must substitute a name or a value.
`( )`	Parentheses separate names of functions from their arguments.
_	An underscore separates an argument type (left) from an argument name (right).
`[ ]`	Brackets indicate that the enclosed argument is optional.

`=>`	A right arrow points to the return values of the function. Also used in code examples in SKILL manuals.
`...`	Three dots indicate that the preceding item can appear any number of times.

Example 2

```
needNCells(
s_cellType | st_userType
x_cellCount
)
=> t/nil
```

Example 2 illustrates two additional syntax characters.

`|`	Vertical bars separate a choice of required options.
`/`	Slashes separate possible return values.

Form Examples

Each form shows you the system defaults:

- Filled-in buttons are the default selections.
- Filled-in values are the default values.

Help and Support Facilities

The following help and support facilities are available as Help menu options:

Help Menu Option	Description
Contents	Invokes Cadence Help with the Virtuoso Analog Design Environment XL User Guide table of contents on display.
What's New	Opens up the Virtuoso What's New document in Cadence Help at the Virtuoso Analog Design Environment XL section.

Help Menu Option	Description
Known Problems and Solutions	Opens up the Virtuoso Known Problems and Solutions document in Cadence Help at the Virtuoso Analog Design Environment XL section.
Virtuoso Documentation	Opens up Cadence Help, initially by default at the Virtuoso Platform What's New overview. To view the entire Virtuoso documentation library contents, if not already on display, select View - Show Navigation.
Cadence Video Library	Opens up Cadence Online Support (COS), using your default web browser, initially displaying the Cadence Video Library site. **Note:** You are required to have a Cadence Online Support account to access these materials.
Cadence Online Support	Displays the Cadence customer support site (COS) on your default web browser. **Note:** You are required to have a Cadence Online Support account to access these materials.
Cadence Users Forum	Displays the Cadence online users forum in your default web browser.
About Analog Design Environment	Displays version and copyright information for Virtuoso Analog Design Environment XL.

1

Getting Started in the Environment

The Virtuoso Analog Design Environment XL is an advanced design and simulation environment that allows you to do the following:

- Draw your schematic using the Virtuoso Schematic Editor.

  You can use the schematic editor to build a circuit or system using high-level functional blocks (such as analog macromodels), gradually filling in the details of these blocks using circuit elements (transistor-level models). The trade-off is speed versus accuracy: High-level functional block simulation is fast; transistor-level simulation is more accurate.

- Define and place design variables (including equations) on your schematic elements

  The simulator evaluates your variables and equations, automatically passing modified values down through the schematic hierarchy. You can archive important aspects of your design with your schematic, which can contain both design equations and circuit topology details.

- Annotate your schematic with DC voltages and transistor operating point information
- Specify device constraints and commands to control your simulation environment
- Simulate interactively such that you can draw, change, and analyze your design; display and manipulate your simulation results; interrupt the simulation so that you can probe through your design hierarchy to check node voltages and currents, then continue your simulation

  The environment supports the following Cadence circuit simulators:

  - Accelerated Parallel Simulator
  - Spectre
  - SpectreRF
  - UltraSim
  - AMS Designer
  - SpectreVerilog (IC6.1.6 ONLY)

- ❑ UltraSimVerilog (IC6.1.6 ONLY)

Alternatively, you can integrate your own third-party simulator using the Open Analog Simulator Interface Socket (OASIS) Direct. See your local Cadence representative for more information and to request a copy of the *OASIS Direct Integrator's Guide*.

Important

The SpectreVerilog and UltrasimVerilog circuit simulators are available only in IC6.1.6 release.

- Set up and run advanced analyses such as
  - ❑ Parametric sweeps
  - ❑ Corners analyses
  - ❑ Monte Carlo statistical analyses
  - ❑ Combination sweeps
- View waveforms and perform post-processing tasks on your simulation results using the Virtuoso® Visualization and Analysis XL tool
- Generate data sheets and specification sheets
- View and manipulate your active setup as well as checkpoints from previous runs
- Backannotate parameter values
- Integrate third-party calculators and waveform viewers into your ADE XL environment
- Customize your user interface (see the *Virtuoso Design Environment User Guide*)

Getting Started in the ADE XL Environment

You can launch ADE XL from the Virtuoso Design Environment Command Interpreter Window (CIW) or from your schematic editing window.

The ADE XL environment consists of a set of menus, toolbars and assistant panes that make up your workspace. You can load a Cadence workspace or create and load a custom workspace. You can specify what workspace to load for a given cellview. For more information about workspaces, see Getting Started with Workspaces in the *Virtuoso Design Environment User Guide*.

See the following topics for more information about the ADE XL environment:

One you have created a test setup cellview, you can open it in the usual ways. See “Working with Cellviews” in the *Virtuoso Design Environment User Guide* for more information.

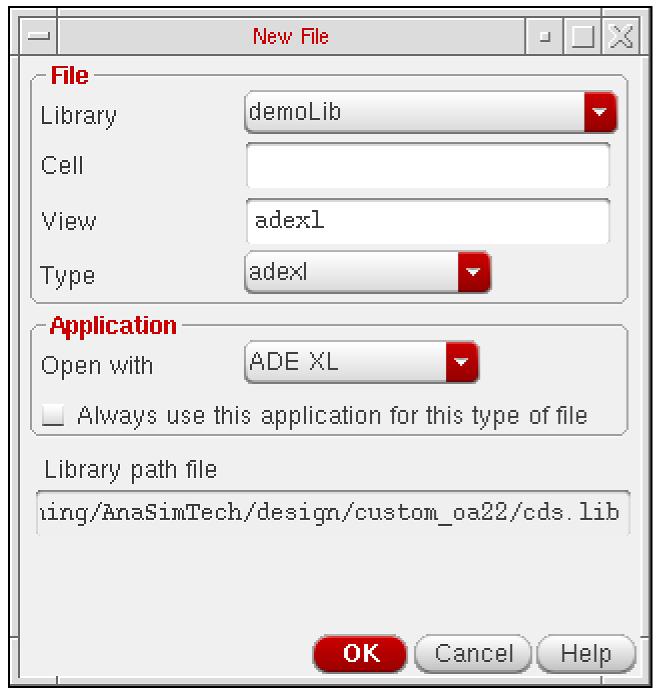
Launching Analog Design Environment XL

To start the environment, start the Virtuoso Design Environment and do one of the following:

To open the environment from the CIW and create a new cellview

1. In the CIW, choose *File – New – Cellview*.

   The New File form appears.



2. In the *Library Name* field, select a library.
3. In the *Cell Name* field, type a new or existing cell name.
4. In the *View Name* field, type a name for your cellview.
5. In the *Type* field, select *adexl*.
6. Click *OK*.

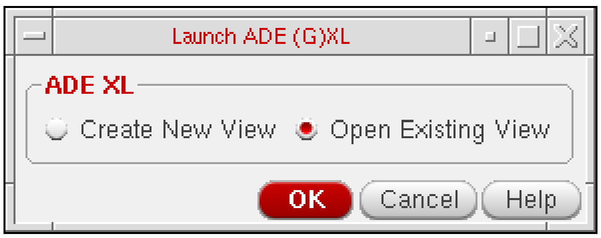
   The environment appears.

<table>
<tr><td>To open an existing ADE XL cellview from the CIW</td><td>1. In the CIW, choose File – Open.
The Open File form appears.
2. Select an ADE XL cellview.
3. Click OK.</td></tr>
<tr><td>To open the environment from the schematic and create a new cellview</td><td>1. In the schematic editing window, choose Launch – ADE XL.
The ADE (G)XL Launch form appears.
2. Perform the procedure described in “Creating a New Setup” on page 37.
Note: If you have descended into a design hierarchy, the environment returns you to the top level of your design when you choose Launch – ADE XL.</td></tr>
<tr><td>To open an existing ADE XL cellview from the schematic</td><td>1. In the schematic editing window, choose Launch – ADE XL.
The ADE (G)XL Launch form appears.
2. Perform the procedure described in “Opening an Existing Setup” on page 39.
Note: If you have descended into a design hierarchy, the environment returns you to the top level of your design when you choose Launch – ADE XL.</td></tr>
<tr><td>To open the environment from the CIW without first selecting a design</td><td>➤ In the CIW, choose Tools – ADE XL.
The ADE (G)XL Launch form appears. From this form, you can create a new test setup or open an existing test setup.
See “Using the ADE (G)XL Launch Form” on page 36 for more information.</td></tr>
</table>

Using the ADE (G)XL Launch Form

The ADE (G)XL Launch form appears when you do one of the following:

- Choose *Tools – ADE XL* in your Command Interpreter Window (CIW).
- Choose *Launch – ADE XL* in your schematic editing window.



From this form, you can create a new test setup or open an existing test setup. See the following sections for information about getting started with any of these tasks:

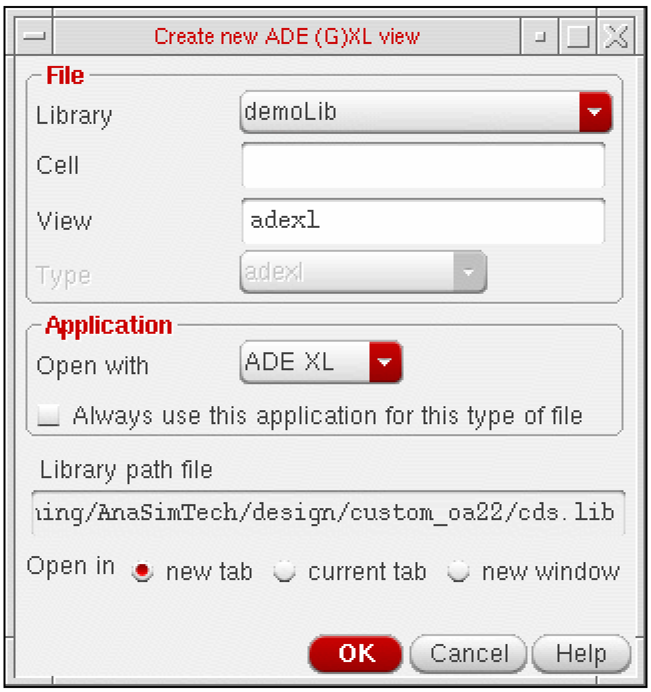
- Creating a New Setup on page 37
- Opening an Existing Setup on page 39

Creating a New Setup

To create a new setup, do the following:

1. On the ADE (G)XL Launch form, select *Create New View*.
2. Click *OK*.

   The Create New ADE (G)XL View form appears.



3. In the *Library Name* drop-down list, select a library.
4. In the *Cell Name* field, type a cell name for your testbench.
5. In the *View Name* field, type a view name for your test setup.

   The default ADE XL view name and view type is `adexl`. For more information about the ADE XL view, see ADE XL View Directory Structure on page 762.
6. Verify that *ADE XL* appears in the *Open with* drop-down list.
7. (Optional) Select the *Always use this application for this view type* check box if you want the program to use ADE XL when opening a view that is the same as what you specified in the *View Name* field.

8. In the *Open in* field, do one of the following:

Select	To
new tab	Open the cellview in a new tab in the schematic editing window. **Note:** If a schematic editing window is not open, ADE XL is opened as a main application in a new window.
current tab	Open the cellview in the current tab in the schematic editing window. **Note:** If a schematic editing window is not open, ADE XL is opened as a main application in a new window.
new window	Open the cellview in a new window. ADE XL is opened as a main application in a new window.

9. Click *OK*.

   The ADE XL environment appears. For more information, see ADE XL Environment at Startup on page 41 and ADE XL Environment User Interface on page 45.

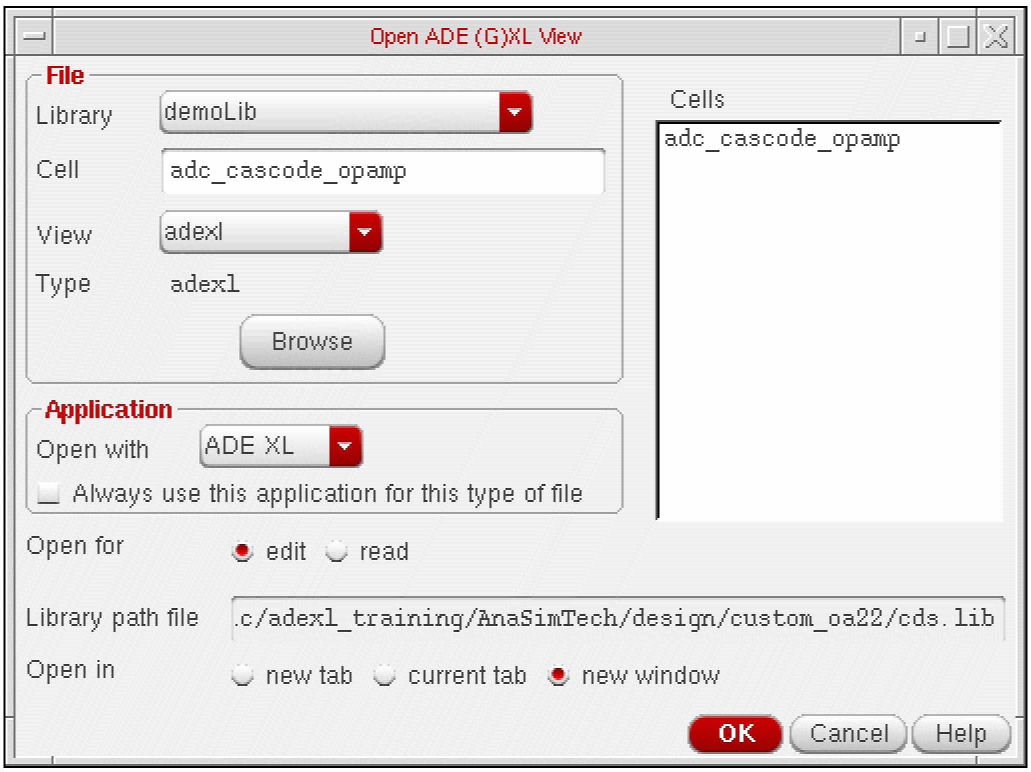
   You can begin specifying your tests and analyses. See Chapter 2, “Specifying Tests and Analyses” for more information.

Opening an Existing Setup

To open an existing setup, do the following:

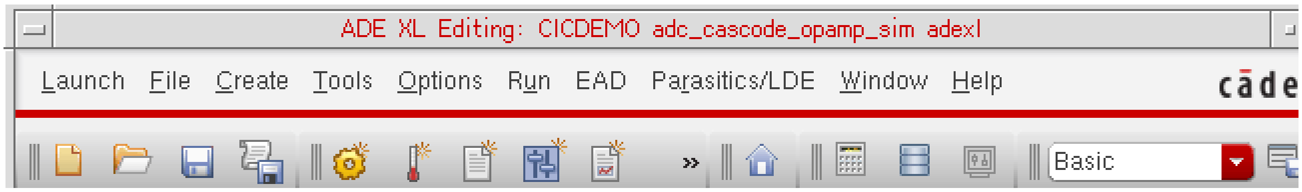
1. On the ADE (G)XL Launch form, select *Open Existing View*.
2. Click *OK*.

   The Open ADE (G)XL View form appears.

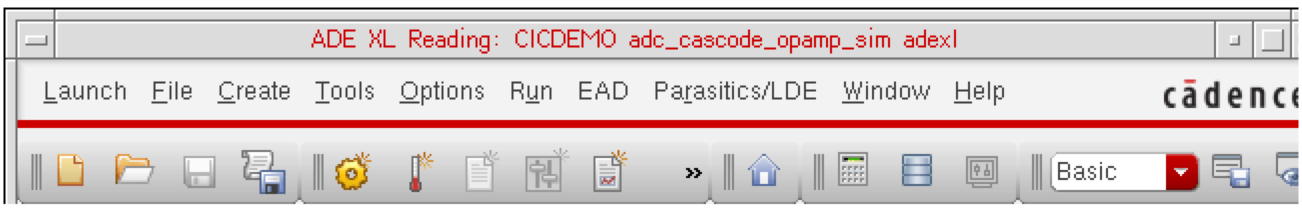


3. Use the *Library Name*, *Cell Name*, and *View Name* fields to select your setup.
4. Verify that *ADE XL* appears in the *Open with* drop-down list.
5. (Optional) Select the *Always use this application for this view type* check box if you want the program to use ADE XL when opening a view that is the same as what you specified in the *View Name* field.
6. (Optional) The *Open for* field indicates that the ADE XL view will be opened in edit mode by default. Select *read* to open the ADE XL view in read-only mode.

The ADE XL title bar displays the text *Editing* if the ADE XL view is opened in edit mode.



The ADE XL title bar displays the text *Reading* if the view is opened in read-only mode. For more information about working with ADE XL views in read-only mode, see Working with Read-Only ADE XL Views on page 775.



7. In the *Open in* field, do one of the following:

Select	To
new tab	Open the cellview in a new tab in the schematic editing window. **Note:** If a schematic editing window is not open, ADE XL is opened as a main application in a new window.
current tab	Open the cellview in the current tab in the schematic editing window. **Note:** If a schematic editing window is not open, ADE XL is opened as a main application in a new window.
new window	Open the cellview in a new window. ADE XL is opened as a main application in a new window.

8. Click *OK*.

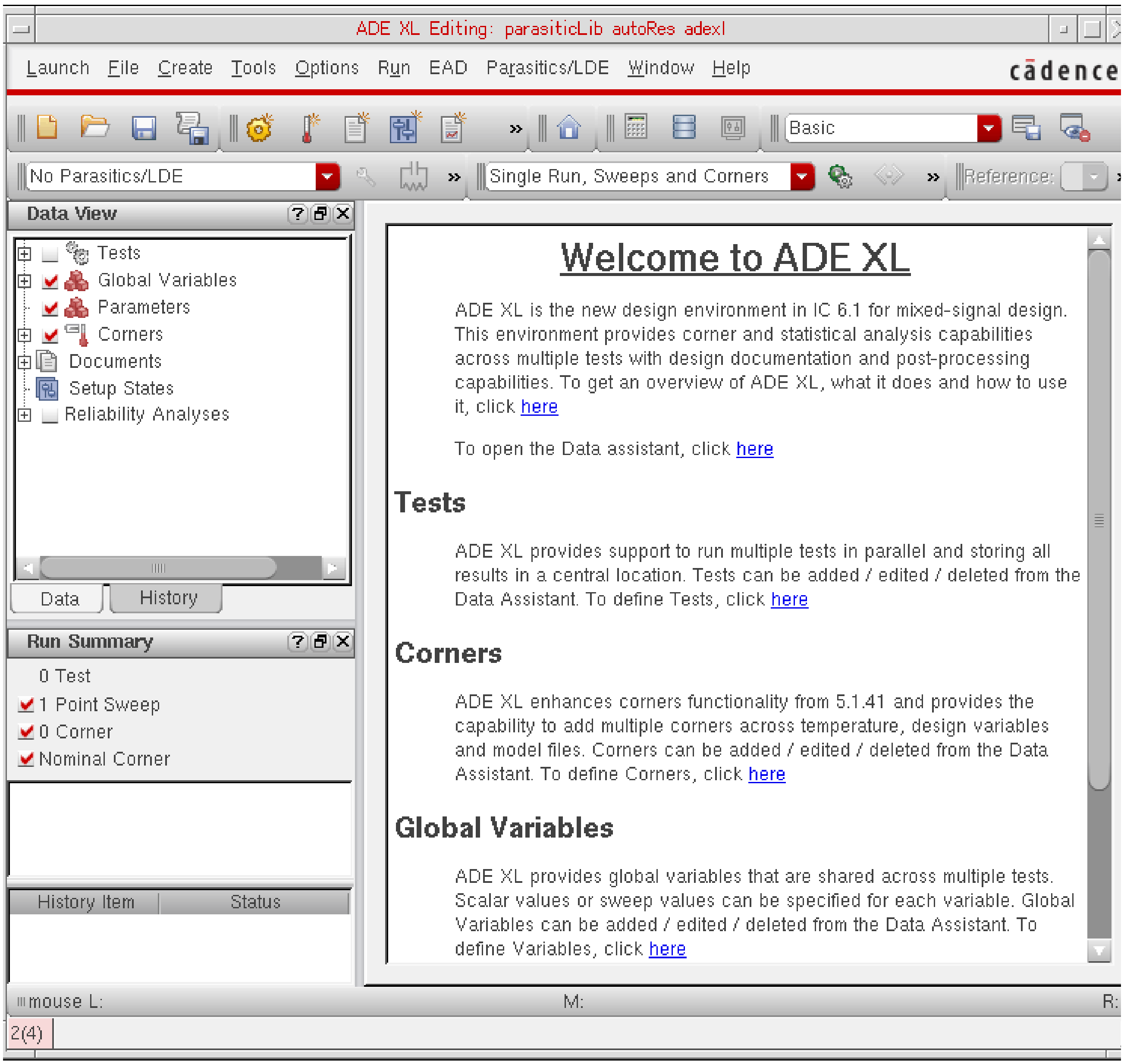
The ADE XL environment appears. For more information, see ADE XL Environment at Startup on page 41 and ADE XL Environment User Interface on page 45.

You can continue specifying tests and analyses. See Chapter 2, "Specifying Tests and Analyses" for more information.

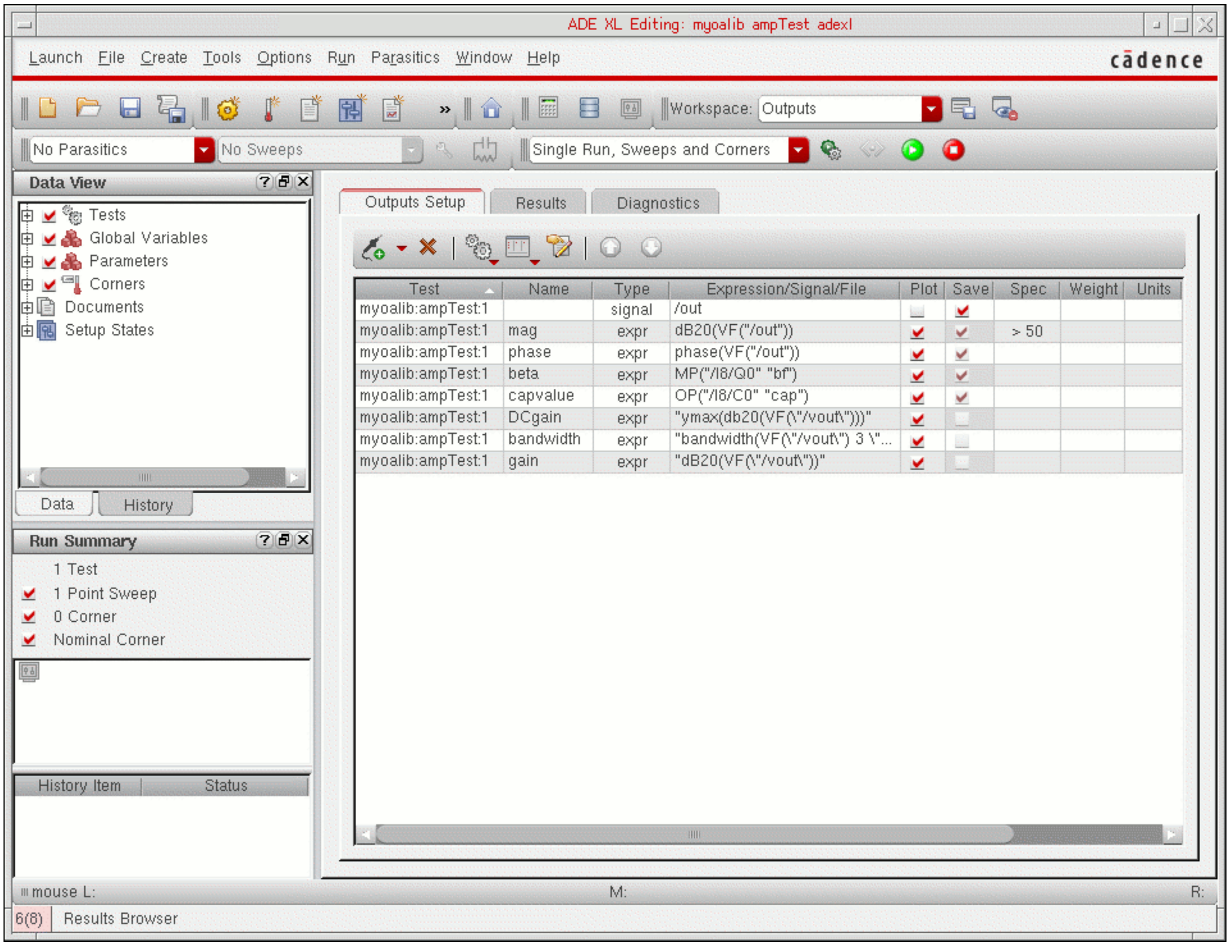
ADE XL Environment at Startup

This section describes how the ADE XL environment appears on startup.

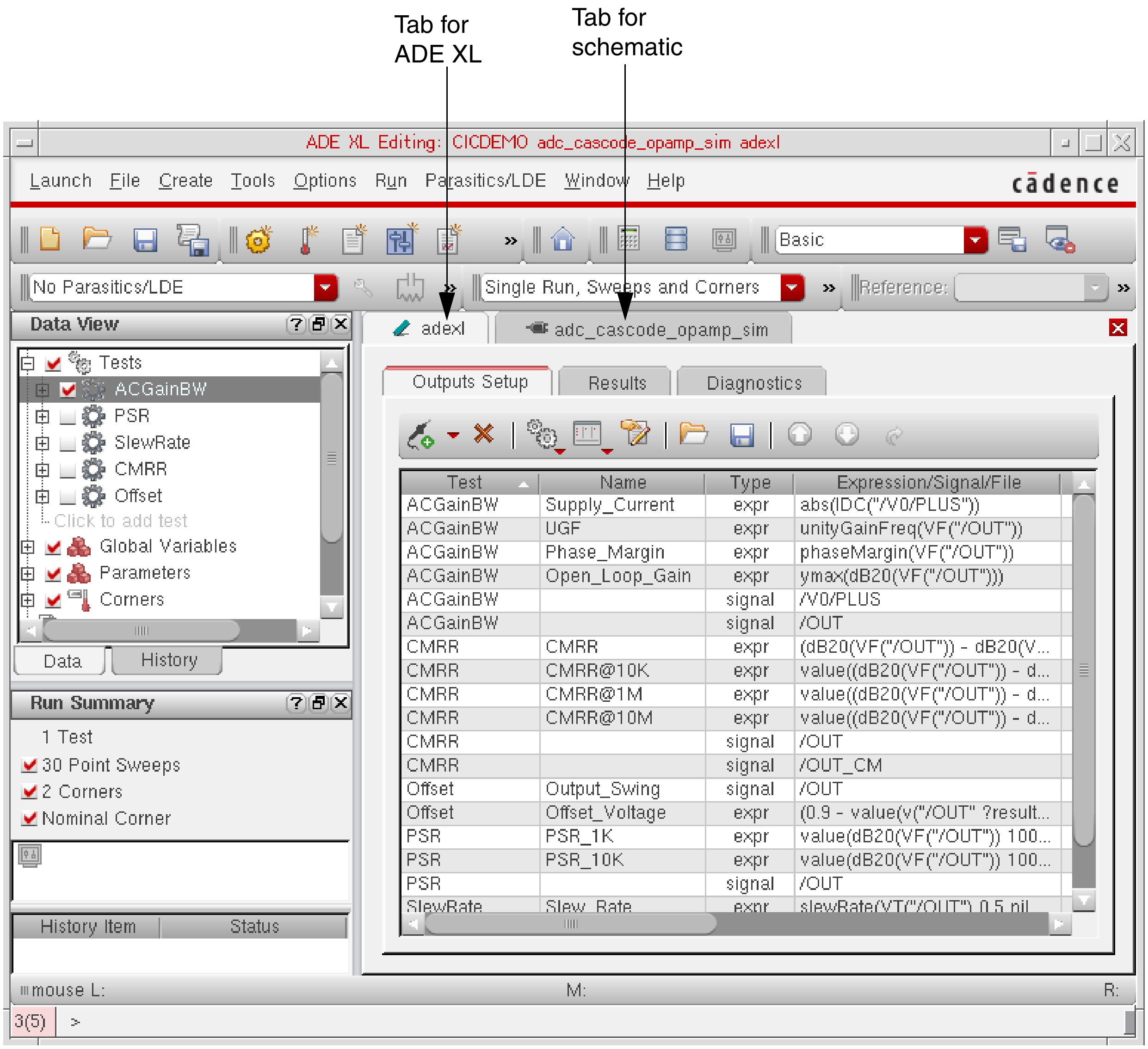
- The Welcome to ADE XL start page is displayed if you created a new setup using the procedure described in Creating a New Setup on page 37. For more information about the Welcome to ADE XL start page, see ADE XL Start Page on page 46.



- The Outputs pane is displayed if you opened an existing setup using the procedure described in Opening an Existing Setup on page 39.

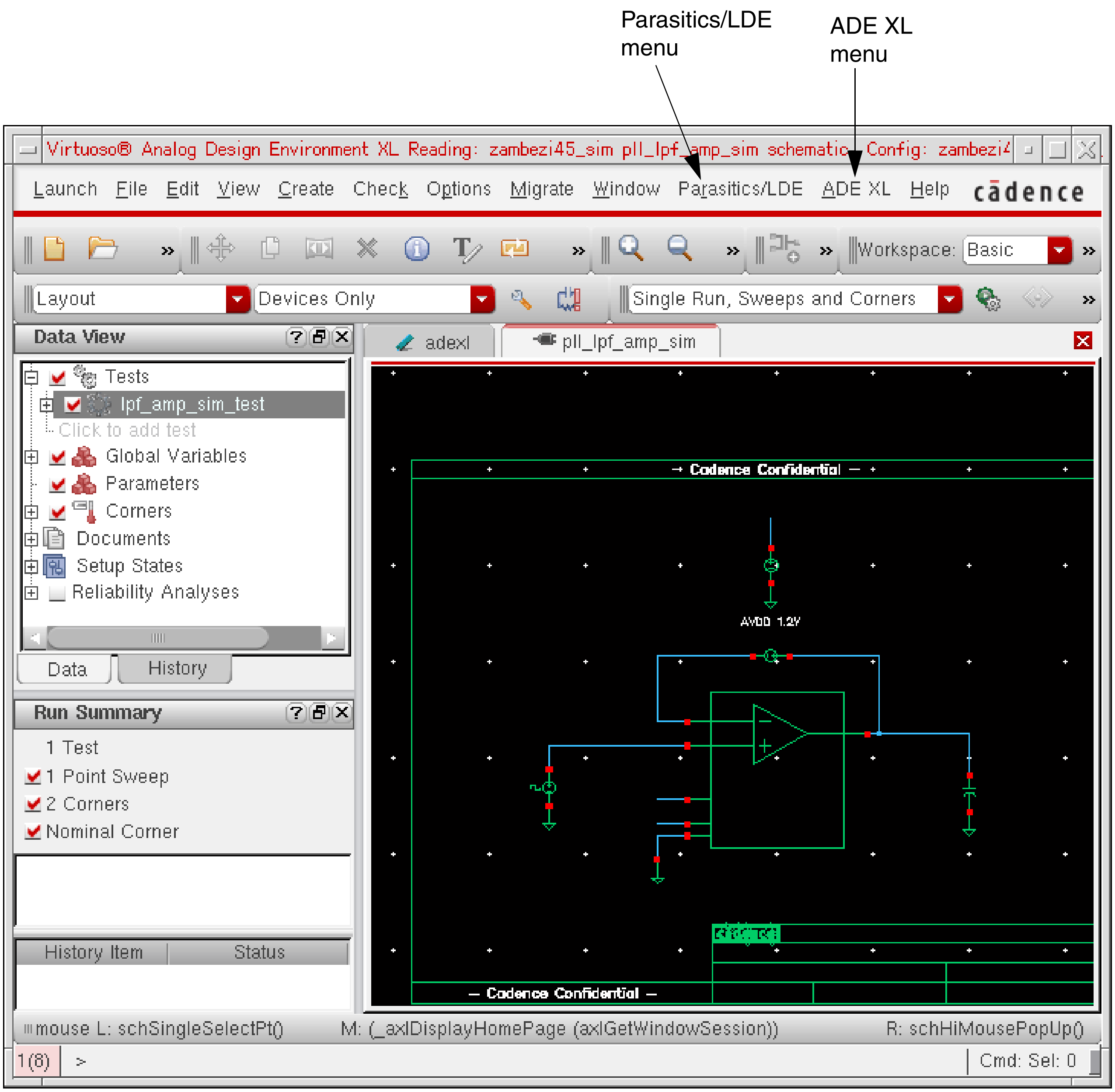


- The ADE XL environment is displayed in a new tab if you choose *Launch – ADE XL* in your schematic editing window and created a new setup or opened an existing setup using the procedures described in Creating a New Setup on page 37 or Opening an Existing Setup on page 39.



The tab for the schematic design displays the name of the cell for the schematic design. The tab for the ADE XL environment displays the name `adexl`. For more information about working with tabs, see the *Virtuoso Design Environment User Guide*.

The *ADE XL* and *Parasitics/LDE* menus appear on the menu banner in the tab for the schematic design. You can access ADE XL related commands from the *ADE XL* menu. For more information, see Menu Bar on page 47. The *Parasitics/LDE* menu lets you access Virtuoso® Parasitic Aware Design to investigate the effects of parasitics on your circuits. For more information, see the *Virtuoso Parasitic Aware Design User Guide*.



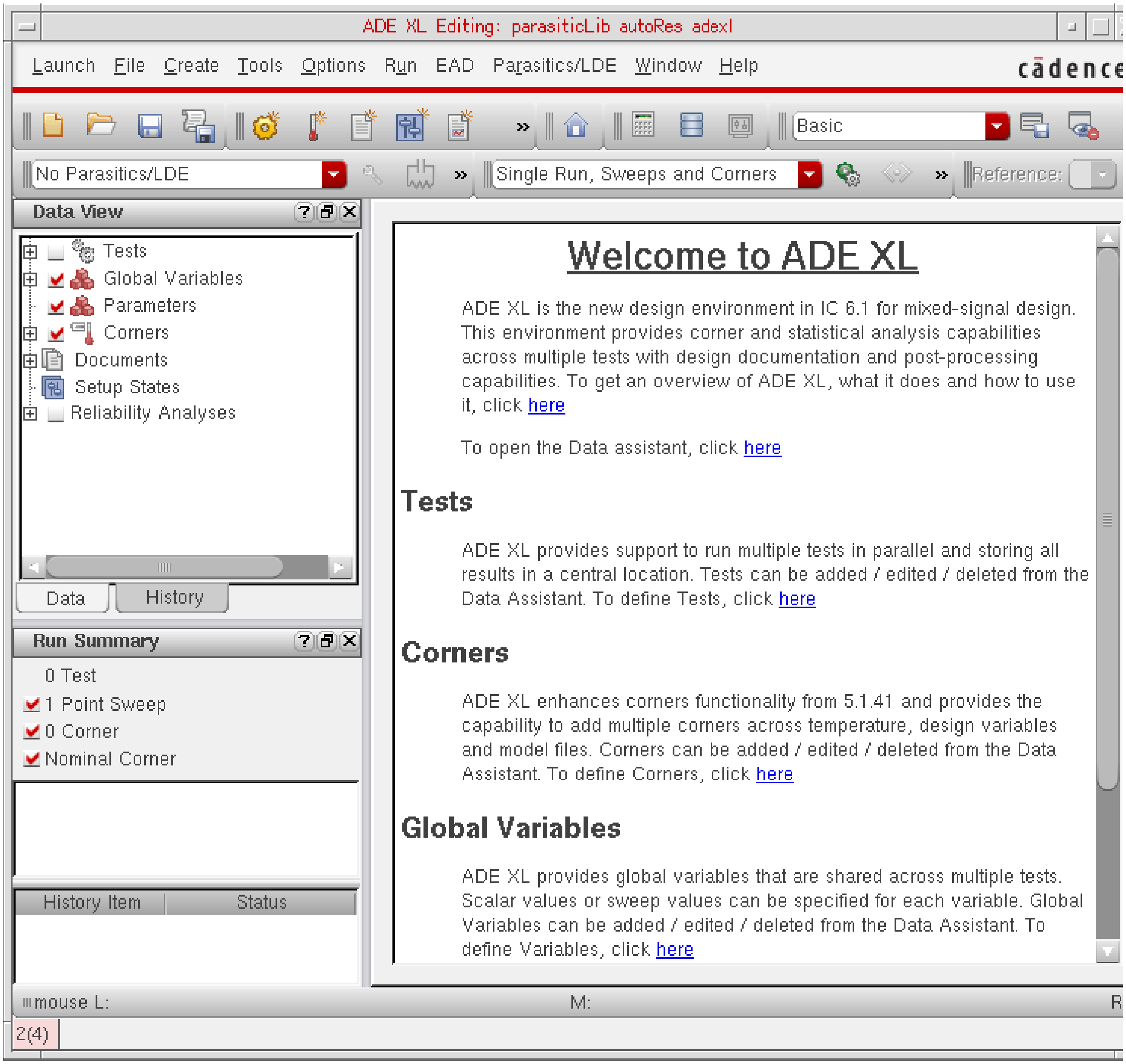
For more information about the user interface in the ADE XL environment, see ADE XL Environment User Interface on page 45.

ADE XL Environment User Interface

This section describes the following parts of the ADE XL environment user interface:

ADE XL Start Page

The Welcome to ADE XL start page provides an overview of ADE XL and guides you through the basic setup tasks in ADE XL. You can click on a hypertext link in the start page to open the form corresponding to a setup task.



The start page is displayed if you do one of the following:

- Create a new setup using the procedure described in Creating a New Setup on page 37.

- Choose *Window – Welcome to ADE XL*.

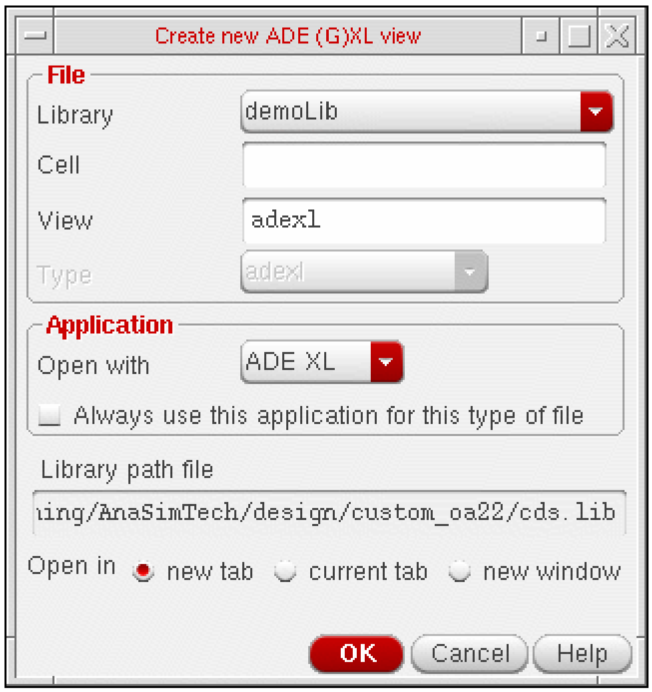
Menu Bar

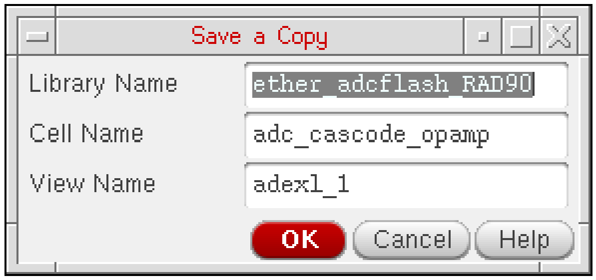
The menu bar in ADE XL has the following menus. You can also access these menus from the *ADE XL* menu that appears on the menu banner in the tab for the schematic design.

- File
- Create
- Tools
- Options
- Run
- EAD
- Parasitics/LDE
- Window
- Help

File

The options in the *File* menu are described below:

File Menu Options	Description
New	Opens the New File form  **Note:** This is the same form that appears when you launch ADE XL from the CIW and create a new cellview.
Open	Opens the Open File form
Save	Saves your ADE XL cellview.

File Menu Options	Description
Save As	Opens the Save a Copy form so you can save a copy of the current ADE XL setup database to a different location.  **Note:** This action does not save history data, only the setup database.
Save Script	Opens the Save OCEAN Script form so you can save simulation setup and conditions to an OCEAN script file.
Save Setup State	Opens the Save Setup State form so you can save a copy of the current ADE XL setup database **Note:** This action does not save history data, only the setup database. For more information, see Creating or Updating a Setup State on page 785.
Load Setup State	Opens the Load Setup State form so you can load an existing setup state. For more information, see Loading a Setup State on page 787.
Remove Setup State	Opens the Remove Setup State form so you can delete an existing setup state. For more information, see Deleting a Setup State on page 790.
Import	Opens the Import Setup form so you can import settings from an existing ADE XL setup database. For more information, see Importing the Simulation Setup on page 778.

File Menu Options	Description
Export	Opens the Export Setup form so you can export settings from from the current ADE XL setup database to an existing ADE XL setup database For more information, see Exporting the Simulation Setup on page 781.
Bookmarks	Allows you to bookmark design views and return to them during the current or future sessions. For information on bookmarks see Using Bookmarks and Views in the *Virtuoso Design Environment User Guide*.
Close	Closes the ADE XL environment

Create

The options in the *Create* menu are described below:

Create Menu Options	Description
Test	Opens the ADE XL Test Editor form so you can add a new test For more information, see Chapter 2, "Specifying Tests and Analyses."
Corner	Opens the Corners Setup form so you can setup corners for simulation For more information, see Chapter 6, "Simulating Corners."
Document	Opens the Choose Documents to be Added form so you can select the documents to be added to the ADE XL view. For more information, see Chapter 18, "Working with Documents."

Create Menu Options	Description
Setup State	Opens the Save Setup State form so you can save a copy of the current ADE XL setup database **Note:** This action does not save history data, only the setup database. For more information, see Creating or Updating a Setup State on page 785.
Datasheet	Opens the Create Datasheet form so you can create a datasheet for the history item selected in the History tab of the Data View assistant pane. For more information, see Creating a Datasheet for a Checkpoint on page 820.
Spec Summary	Opens the Spec Summary form so you can view the specifications summary for the results of a simulation run. For more information, see Working with the Specification Summary on page 749.
Spec Comparison	Opens the Spec Comparison form so you can compare measured results of output expressions for: ■ Any two history items. ■ Any two tests in the same history item or in two different history items. ■ Any two design points in the same history item or in two different history items. For more information, see Comparing Results on page 659.

Tools

The options in the *Tools* menu are described below:

Tools Menu Options	Description
Calculator	Opens the Calculator window
Results Browser	Opens the Results Browser window

Tools Menu Options	Description
Job Monitor	Opens the Job Monitor

Options

The options in the *Options* menu are described below:

Options Menu Options	Description
Job Setup	Opens the Job Policy Setup form
Run Options	Opens the Run Options form
Save	Opens the Save Options form
Outputs Formatting	Opens the Default Formatting Options form
Plotting/Printing	Opens the ADE XL Plotting/Printing Options form

Run

The options in the *Run* menu are described below:

Run Menu Options	Description
Single Run, Sweeps and Corners	Run simulations across multiple tests with sweeps and corners analyses.
Monte Carlo Sampling	Opens the Monte Carlo form for specifying Monte Carlo method and sampling options.
Global Optimization (ADE GXL feature)	Perform global optimization using a parallel simulated annealing algorithm For more information, see the *Virtuoso Analog Design Environment GXL User Guide*.
Local Optimization (ADE GXL feature)	Perform local optimization For more information, see the *Virtuoso Analog Design Environment GXL User Guide*.

Run Menu Options	Description
Improve Yield (ADE GXL feature)	Performs an optimization run to improve the yield of your design. For more information, see the *Virtuoso Analog Design Environment GXL User Guide*.
High Yield Estimation (ADE GXL feature)	Uses worst case distance algorithm to estimate yield for circuits with high yields. Decreases the number of simulations required to estimate yield when compared with Monte Carlo simulation. For more information, see High Yield Estimation in the *Virtuoso Analog Design Environment GXL User Guide*.
Sensitivity Analysis (ADE GXL feature)	Performs sensitivity analysis. For more information, see the *Virtuoso Analog Design Environment GXL User Guide*.
Feasibility Analysis (ADE GXL feature)	Verifies that the circuit topology meets all operating point requirements. For more information, see the *Virtuoso Analog Design Environment GXL User Guide*.
Worst Case Corners (ADE GXL feature)	Uses sensitivity analysis to identify worst case corners for each design specification and adds those corners to the Corners Setup form. For more information, see Creating Worst Case Corners in the *Virtuoso Analog Design Environment GXL User Guide*.
Manual Tuning (ADE GXL feature)	Lets you tune your design by varying the values of parameters, running multiple simulations, and then comparing results. For more information, see Manual Tuning in the *Virtuoso Analog Design Environment GXL User Guide*.
Size Over Corners (ADE GXL feature)	Optimizes test benches over a large number of corners. For more information, see the *Virtuoso Analog Design Environment GXL User Guide*.
Edit Reference Point	Used to specify reference points for global variables and device parameters. For more information, see "Creating, Viewing, and Modifying Reference Points" in the *Virtuoso Analog Design Environment GXL User Guide*.

Run Menu Options	Description
Submit Point	Submits a design point from the history results for evaluation. For more information, see Submitting a Point.
Stop Simulation	Stops the simulation run.
Stop All Jobs	Stops all jobs you started during the current session regardless of their current state (started, getting configured, running). *Important* Use this option with extreme caution.

EAD

The EAD menu lets create setup and save data for the Electrically Aware Design flow in Virtuoso. The options available in this menu are described below:

Options Menu Options	Description
Setup	Displays the EAD Setup form. For more information, see Preparing the EAD Setup for Simulation in *Virtuoso Electrically Aware Design Flow Guide*.
Signal Selection	Opens the design schematic in a new tab and displays the Parasitics & Electrical Setup assistant to enable selection of signals for the EAD flow. For more information, see Selecting Signals to Save Currents in *Virtuoso Electrically Aware Design Flow Guide*.

Parasitics/LDE

The *Parasitics/LDE* menu lets you investigate the effects of parasitics and Layout Dependent Effects (LDEs) on your circuits. You can report on parasitics that exist in your design, show or hide them on your design, and create refined extracted cellviews.

For more information, see the following links:

- *Virtuoso Parasitic Aware Design User Guide*

- Simulating Designs with Layout-Dependent Effects (LDEs)

Window

The options in the *Window* menu are described below:

Options Menu Options	Description
Welcome to ADE XL	Displays the Welcome to ADE XL start page. For more information, see ADE XL Start Page on page 46.
Assistants	Displays or hides the selected assistant pane For more information about assistant panes, see the *Virtuoso Design Environment User Guide*.
Workspaces	Displays the selected workspace For more information about workspaces, see "Getting Started with Workspaces" in the *Virtuoso Design Environment User Guide*.
Toolbars	Displays or hides the selected toolbar You can also customize toolbars to regroup the commands as required. For more information about customizing toolbars, see the *Virtuoso Design Environment User Guide*.
Tabs	Allows you to work with tabs in the session window For more information about working with tabs, see the *Virtuoso Design Environment User Guide*.

Help

For information about the options in the *Help* menu, see Help and Support Facilities on page 28.

Toolbars

ADE XL has the following toolbars:

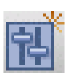
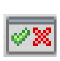
- File

- Create
- Browse
- Go
- Parasitic Mode
- Reference History
- Run
- Setup
- Tools
- Workspaces

File

Icon	Name	Description
	New	Opens the New File form
	Open	Opens the Open File form
	Save	Saves your ADE XL cellview
	Save Script As	Opens the Save OCEAN Script form so you can save simulation setup and conditions to an OCEAN script file.

Create

Icon	Name	Description
	Create Test	Opens the ADE XL Test Editor form so you can add a new test For more information, see Chapter 2, "Specifying Tests and Analyses."
	Create Corner	Opens the Corners Setup form so you can setup corners for simulation For more information, see Chapter 6, "Simulating Corners."
	Create Document	Opens the Choose Documents to be Added form so you can select the documents to be added to the ADE XL view. For more information, see Chapter 18, "Working with Documents."
	Create Setup State	Opens the Save Setup State form so you can save a copy of the current ADE XL setup database **Note:** This action does not save history data, only the setup database. For more information, see Creating or Updating a Setup State on page 785.
	Create Datasheet	Opens the Create Datasheet form so you can create a datasheet for the history item selected in the History tab of the Data View assistant pane For more information, see Creating a Datasheet for a Checkpoint on page 820.
	Create Spec Summary	Opens the Spec Summary form so you can view the specifications summary for the results of a simulation run. For more information, see Working with the Specification Summary on page 749.

Icon	Name	Description
	Create Spec Comparison	Opens the Spec Comparison form so you can compare measured results of output expressions for: ■ Any two history items. ■ Any two tests in the same history item or in two different history items. ■ Any two design points in the same history item or in two different history items. For more information, see Comparing Results on page 659.

Browse

Icon	Name	Description
	Home	Displays the Welcome to ADE XL start page. For more information, see ADE XL Start Page on page 46.

Go

The Go toolbar allows you to do the following:

- Sequentially or non-sequentially navigate through a cellview hierarchy.
- Navigate between cells and views in various designs.

For more information about the Go toolbar, see the *Virtuoso Design Environment User Guide*.

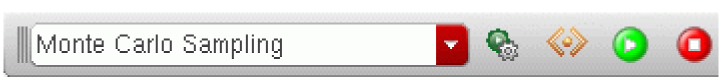
Parasitic Mode

For more information about the Parasitics Mode toolbar, see the *Virtuoso Parasitic Estimation and Analysis User Guide*.

Reference History

Icon	Name	Description
	Reference drop-down list	Lets you select the reference history item for incremental simulation runs. For more information, see Running an Incremental Simulation on page 497.
	Show Differences	Opens the Comparison: Active Setup v/s Reference History form.
	Reference History Options	Opens the Reference History form.

Run



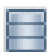
Icon	Name	Description
	Select a Run Mode	Specifies the run mode for simulation For more information, see Specifying the Run Mode on page 71.

Icon	Name	Description
	Simulation Options	Opens a form for specifying the simulation options for the following run modes selected in the *Select a Run Mode* drop-down list: ■ Monte Carlo Sampling ■ Global Optimization ■ Local Optimization ■ Improve Yield
	Edit Reference Point	Opens the Edit Reference Point form for specifying a reference point for Improve Yield, Global Optimization, Monte Carlo Sampling and Sensitivity Analysis runs.
	Run Simulation	Runs the simulation for the run mode selected in the *Select a Run Mode* drop-down list
	Stop Simulation	Stops the simulation run

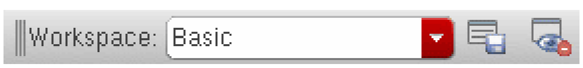
Setup

Icon	Name	Description
	Load	Opens the Load Setup State form so you can load an existing setup state. For more information, see Loading a Setup State on page 787.
	Save Setup	Opens the Save Setup State form so you can save a copy of the current ADE XL setup database **Note:** This action does not save history data, only the setup database. For more information, see Loading a Setup State on page 787.

Tools

Icon	Name	Description
	Calculator	Opens the Calculator window
	Results Browser	Opens the Results Browser window
	Job Monitor	Opens the Job Monitor

Workspaces



The Workspaces toolbar allows you work with workspaces. For more information, see "Getting Started with Workspaces" in the *Virtuoso Design Environment User Guide*.

Assistant Panes

The following assistant panes are available in ADE XL:

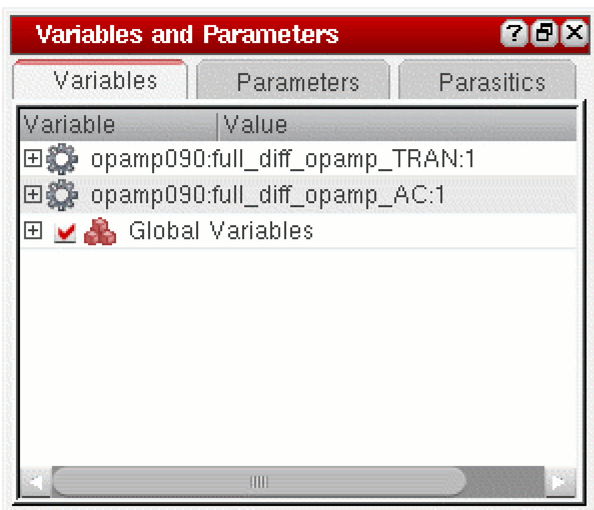
- Variables and Parameters
- Run Summary
- Setup DB Viewer
- Data View
- Variable Display
- Parasitic Filters
- Parasitic Estimates
- Compare Parasitics Report

Variables and Parameters

The Variables and Parameters assistant pane allows you to view and set up design variables, global variables and parameters.

To display the Variables and Parameters assistant pane, do the following:

➤ Choose *Window – Assistant – Variables and Parameters*.



For information about using the Variables and Parameters assistant pane, see the following:

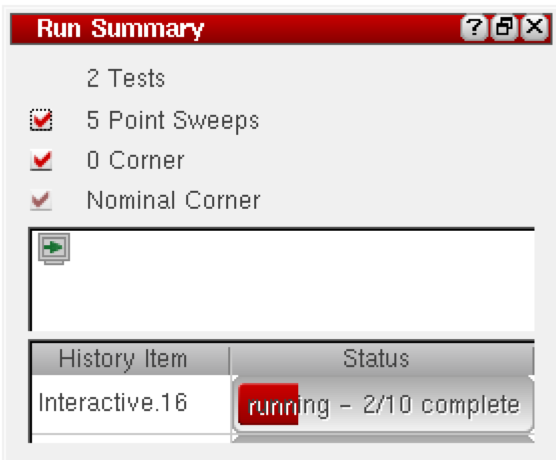
To use the Variables and Parameters pane to work with	See
Design variables, global variables and instance parameters	Chapter 4, "Working with Global Variables" Chapter 3, "Working with Design Variables and Instance Parameters"
Matched parameter constraints	Chapter 5, "Working with Constraints"
Parasitic parameters	*Virtuoso Parasitic Estimation and Analysis User Guide*. **Note:** The Parasitics tab is displayed only if your design has swept parasitic parameters.

Run Summary

The Run Summary assistant pane displays a summary of the simulation setup and the status of simulation runs.

To display the Run Summary assistant pane, do the following:

➤ Choose *Window – Assistant – Run Summary*.



For information about using the Run Summary assistant pane, see the following:

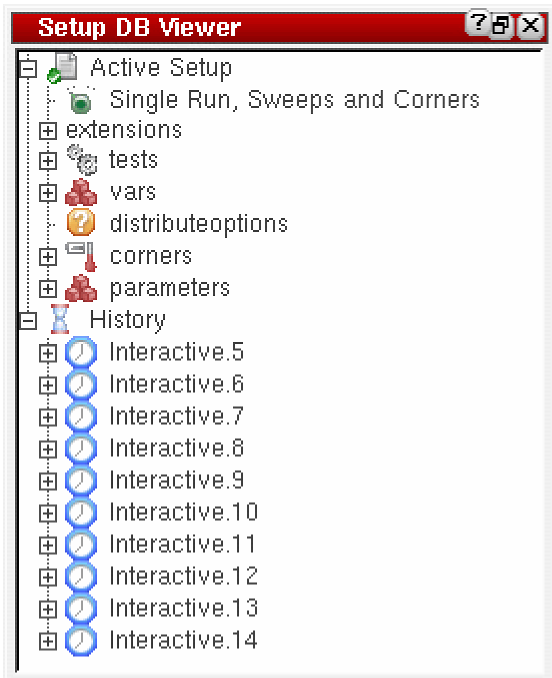
- Disabling and Enabling All Point Sweeps
- Disabling and Enabling All Corners
- Disabling and Enabling the Nominal Corner on page 243
- Viewing Job Status on page 482

Setup DB Viewer

The Setup DB Viewer assistant pane provides a graphical view of the ADE XL setup database.

To display the Setup DB Viewer assistant pane, do the following:

➤ Choose *Window – Assistant – Setup DB Viewer*.



For information about using the Setup DB Viewer assistant pane, see Viewing Active Setup Details on page 810.

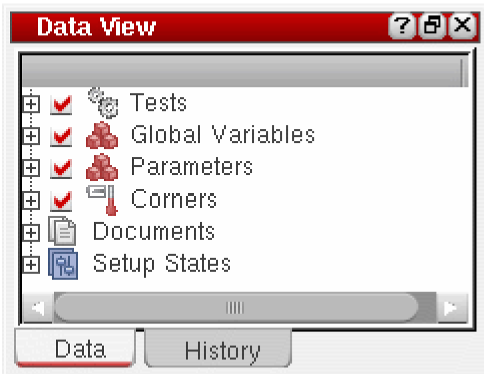
Data View

The Data View assistant pane provides a single user interface to quickly view and set up commonly used setup data.

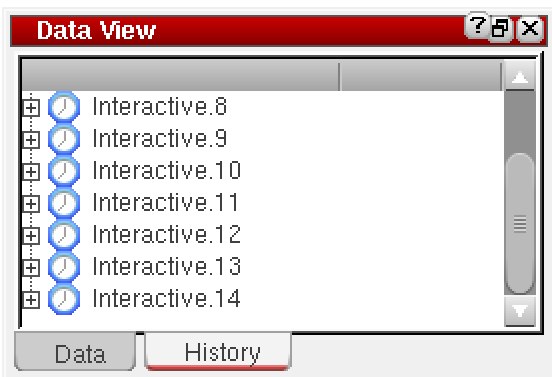
To display the Data View assistant pane, do the following:

➤ Choose *Window – Assistant – Data View*.

The Data tab on the Data View assistant pane allows you to quickly view and set up tests, global variables, parameters and corners, and manage documents and setup states.



The History tab on the Data View assistant pane allows you to work with history items.



For information about using the Data View assistant pane, see the following:

To use the Data View assistant pane to work with	See
Tests	Chapter 2, "Specifying Tests and Analyses"
Design variables, global variables and instance parameters	Chapter 4, "Working with Global Variables" Chapter 3, "Working with Design Variables and Instance Parameters"
Matched parameter constraints	Chapter 5, "Working with Constraints"

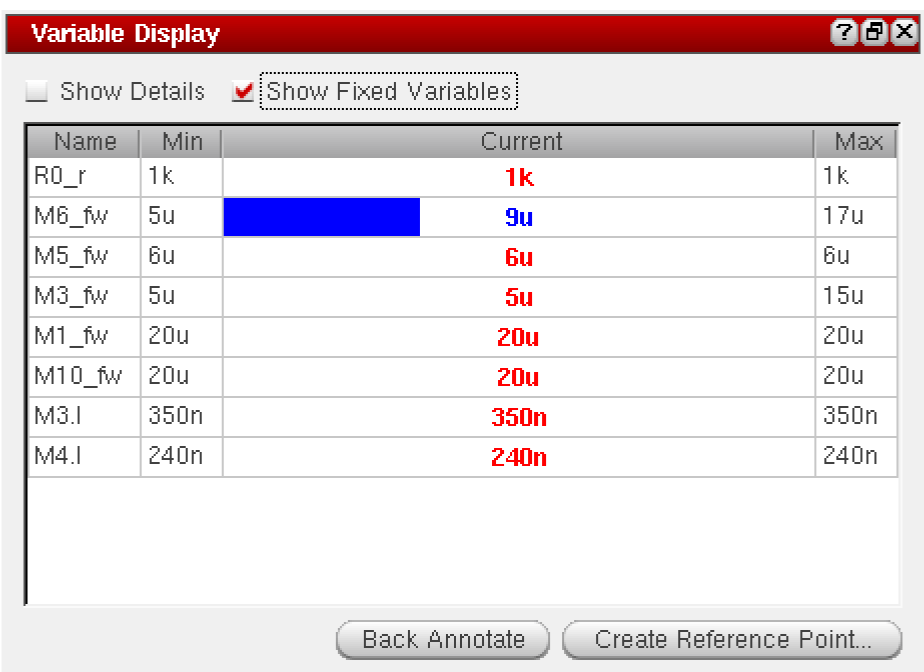
To use the Data View assistant pane to work with	See
Corners	Chapter 6, "Simulating Corners"
History Items	Chapter 17, "Working with Checkpoints"
Documents	Chapter 18, "Working with Documents"
Setup States	Chapter 16, "Working with the Simulation Setup"

Variable Display

The Variable Display assistant pane displays the current, minimum and maximum value set for each variable and parameter for the selected design point.

To display the Variable Display assistant pane, do the following:

➤ Choose *Window – Assistant – Variable Display*.



For more information about the Variable Display assistant pane, see Showing Variables for a Design Point on page 570.

Parasitic Filters

For more information about the Parasitic Filters assistant pane, see the *Virtuoso Parasitic Estimation and Analysis User Guide*.

To display the Parasitic Estimates assistant pane, do the following:

➤ Choose *Window – Assistant – Parasitic Filters*.

Parasitic Estimates

For more information about the Parasitic Estimates assistant pane, see the *Virtuoso Parasitic Estimation and Analysis User Guide*.

To display the Parasitic Estimates assistant pane, do the following:

➤ Choose *Window – Assistant – Parasitic Estimates*.

Compare Parasitics Report

For more information about the Parasitics Report assistant pane, see the *Virtuoso Parasitic Estimation and Analysis User Guide*.

To display the Compare Parasitics Report assistant pane, do the following:

➤ Choose *Window – Assistant – Parasitics Report*.

Outputs

The Outputs pane allows you to setup simulation outputs and view results and diagnostics information. This pane contains the following three tabs:

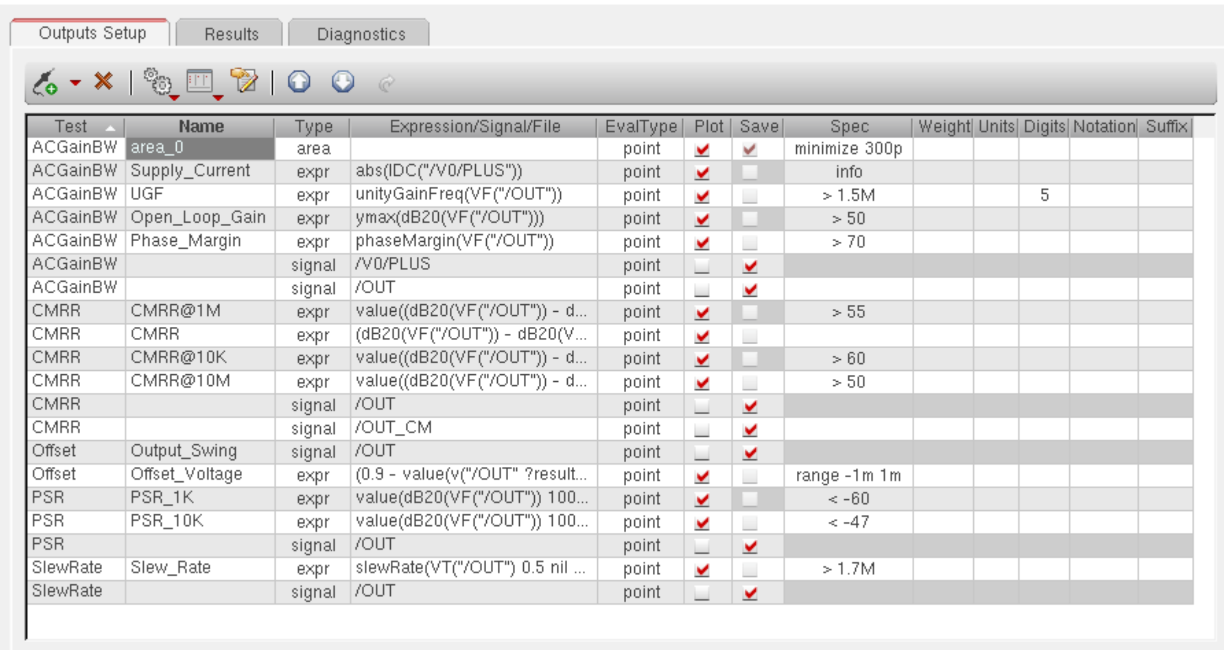
- Outputs Setup
- Results
- Diagnostics

Tip

You can use the `Ctrl + Page Up` or `Ctrl + Page Down` keys to switch between the tabs on the Outputs pane.

Outputs Setup

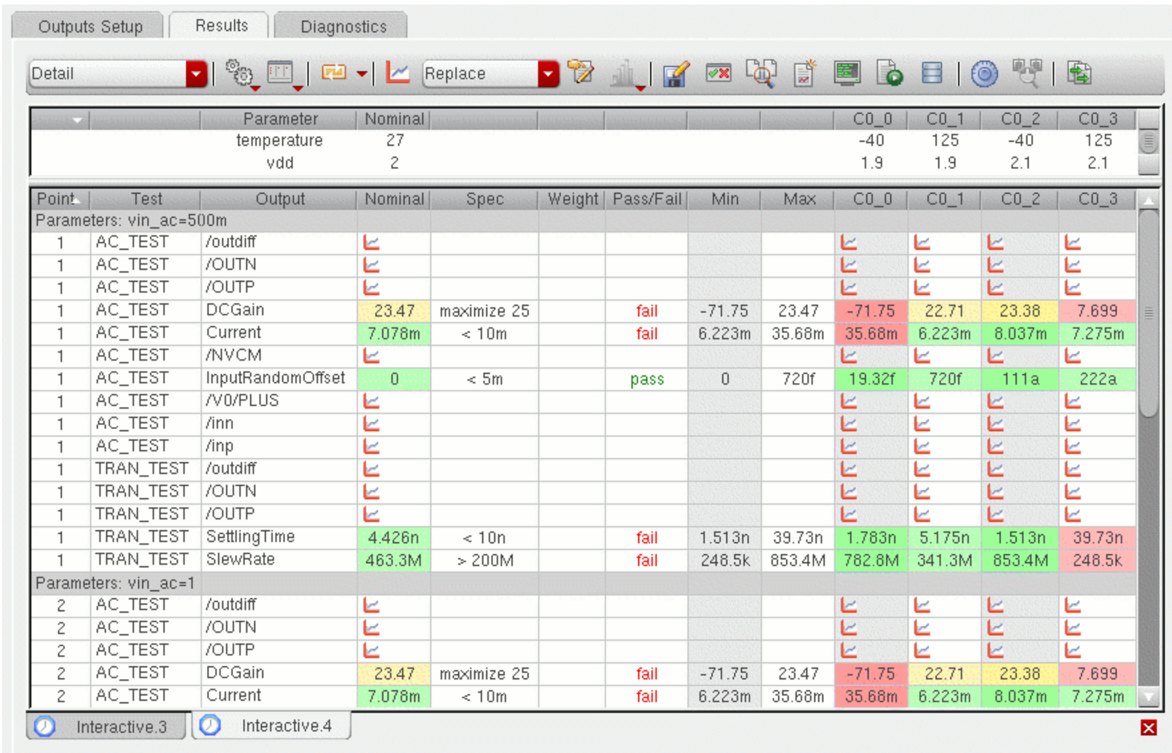
The Outputs Setup tab of the Outputs pane allows you to specify nets, terminals, and measurements you want to save and plot. For more information about using the Outputs Setup tab, see Chapter 11, "Selecting Data to Save and Plot."



Test	Name	Type	Expression/Signal/File	EvalType	Plot	Save	Spec	Weight	Units	Digits	Notation	Suffix
ACGainBW	area_0	area		point	✓	✓	minimize 300p					
ACGainBW	Supply_Current	expr	abs(IDC("/V0/PLUS"))	point	✓		info					
ACGainBW	UGF	expr	unityGainFreq(VF("/OUT"))	point	✓		> 1.5M			5		
ACGainBW	Open_Loop_Gain	expr	ymax(dB20(VF("/OUT")))	point	✓		> 50					
ACGainBW	Phase_Margin	expr	phaseMargin(VF("/OUT"))	point	✓		> 70					
ACGainBW		signal	/V0/PLUS	point		✓						
ACGainBW		signal	/OUT	point		✓						
CMRR	CMRR@1M	expr	value((dB20(VF("/OUT")) - d...	point	✓		> 55					
CMRR	CMRR	expr	(dB20(VF("/OUT")) - dB20(V...	point	✓							
CMRR	CMRR@10K	expr	value((dB20(VF("/OUT")) - d...	point	✓		> 60					
CMRR	CMRR@10M	expr	value((dB20(VF("/OUT")) - d...	point	✓		> 50					
CMRR		signal	/OUT	point		✓						
CMRR		signal	/OUT_CM	point		✓						
Offset	Output_Swing	signal	/OUT	point		✓						
Offset	Offset_Voltage	expr	(0.9 - value(v("/OUT" ?result...	point	✓		range -1m 1m					
PSR	PSR_1K	expr	value(dB20(VF("/OUT")) 100...	point	✓		< -60					
PSR	PSR_10K	expr	value(dB20(VF("/OUT")) 100...	point	✓		< -47					
PSR		signal	/OUT	point		✓						
SlewRate	Slew_Rate	expr	slewRate(VT("/OUT") 0.5 nil ...	point	✓		> 1.7M					
SlewRate		signal	/OUT	point		✓						

Results

The Results tab of the Outputs pane allows you to view simulation results. For more information about using the Results tab, see Chapter 14, “Viewing, Plotting, and Printing Results.”

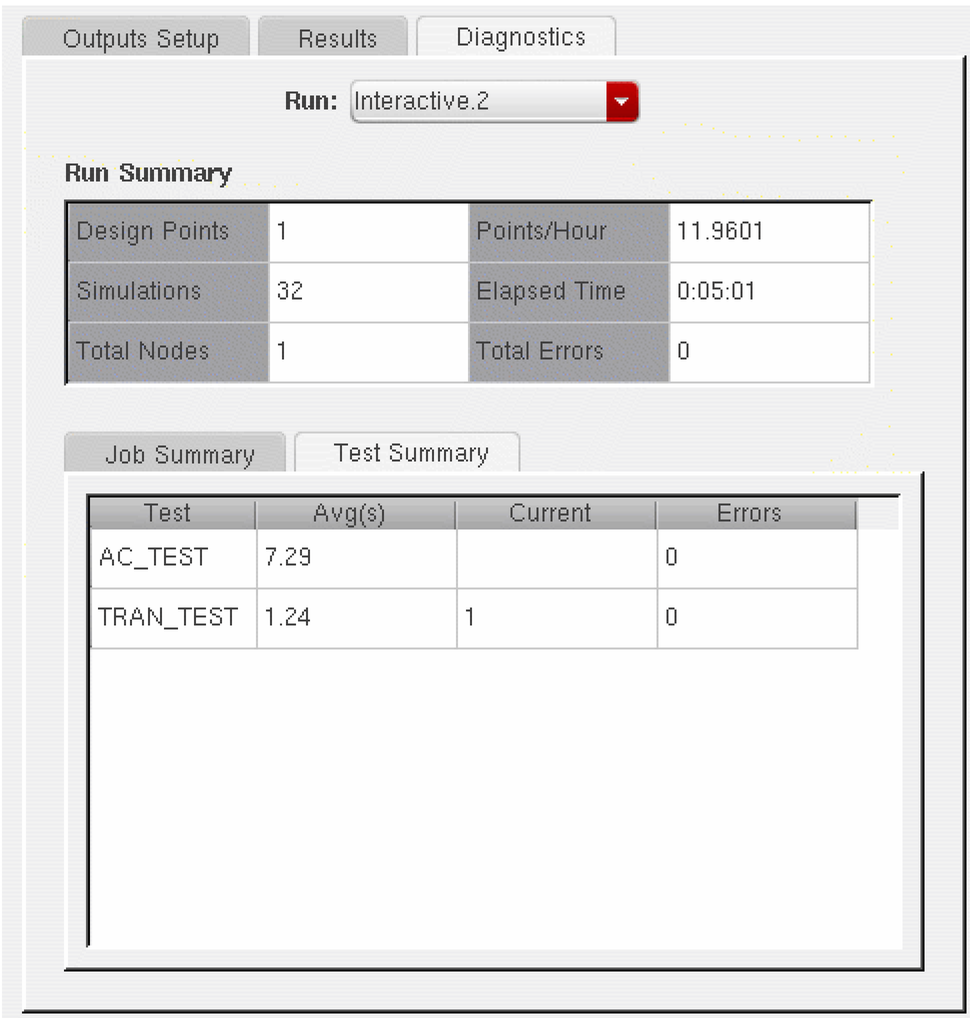


Parameter	Nominal	C0_0	C0_1	C0_2	C0_3
temperature	27	-40	125	-40	125
vdd	2	1.9	1.9	2.1	2.1

Point	Test	Output	Nominal	Spec	Weight	Pass/Fail	Min	Max	C0_0	C0_1	C0_2	C0_3
Parameters: vin_ac=500m												
1	AC_TEST	/outdiff										
1	AC_TEST	/OUTN										
1	AC_TEST	/OUTP										
1	AC_TEST	DCGain	23.47	maximize 25		fail	-71.75	23.47	-71.75	22.71	23.38	7.699
1	AC_TEST	Current	7.078m	< 10m		fail	6.223m	35.68m	35.68m	6.223m	8.037m	7.275m
1	AC_TEST	/NVCM										
1	AC_TEST	InputRandomOffset	0	< 5m		pass	0	720f	19.32f	720f	111a	222a
1	AC_TEST	/V0/PLUS										
1	AC_TEST	/inn										
1	AC_TEST	/inp										
1	TRAN_TEST	/outdiff										
1	TRAN_TEST	/OUTN										
1	TRAN_TEST	/OUTP										
1	TRAN_TEST	SettlingTime	4.426n	< 10n		fail	1.513n	39.73n	1.783n	5.175n	1.513n	39.73n
1	TRAN_TEST	SlewRate	463.3M	> 200M		fail	248.5k	853.4M	782.8M	341.3M	853.4M	248.5k
Parameters: vin_ac=1												
2	AC_TEST	/outdiff										
2	AC_TEST	/OUTN										
2	AC_TEST	/OUTP										
2	AC_TEST	DCGain	23.47	maximize 25		fail	-71.75	23.47	-71.75	22.71	23.38	7.699
2	AC_TEST	Current	7.078m	< 10m		fail	6.223m	35.68m	35.68m	6.223m	8.037m	7.275m

Diagnostics

The Diagnostics tab of the Outputs pane allows you to view runtime and other run-related information during and after a simulation run. For more information about using the Results tab, see Viewing Diagnostics Information on page 494



Workspaces

The following workspaces are available in ADE XL:

- Basic Workspace
- Outputs Workspace

For more information about workspaces, see "Getting Started with Workspaces" in the *Virtuoso Design Environment User Guide*.

Basic Workspace

The following panes appear in the Basic workspace:

- Data View
- Run Summary
- Outputs

Outputs Workspace

The Outputs workspace displays only the Outputs pane.

Specifying the Run Mode

A run mode specifies the type of simulation you want to run in ADE XL.

You can choose a run mode to be used for simulation from the Run menu. Alternatively, you can select a run mode in the *Select a Run Mode* drop-down list.

Simulators Supported for Run Modes

The following simulators are supported for the Single Run, Sweeps and Corners, Global Optimization, Local Optimization, Sensitivity Analysis, Feasibility Analysis and Size Over Corners run modes:

- *spectre* (Virtuoso Spectre circuit simulator)
- *aps* (Virtuoso Accelerated Parallel simulator)
- *UltraSim* (Virtuoso UltraSim simulator)
- *hspiceD* (HSPICE® simulator from Synopsys, Inc)
- *ams* (Virtuoso AMS Designer Simulator)
- *spectreVerilog*
- *UltraSimVerilog*

Important

The SpectreVerilog and UltrasimVerilog circuit simulators are available only in IC6.1.6 release.

The following simulators are supported for the Monte Carlo Sampling and Improve Yield run modes:

- *spectre* (Virtuoso Spectre circuit simulator)
- *aps* (Virtuoso Accelerated Parallel simulator)
- *ams* (Virtuoso AMS Designer simulator with Spectre or APS as the solver.

  Note the following:

  - Ensure that you are using the Cadence IUS 9.2 or later version of Virtuoso AMS Designer simulator.
  - You cannot use Virtuoso AMS Designer simulator with UltraSim as the solver to run Monte Carlo analysis.
  - Running Monte Carlo simulation in interactive mode (using SimVision) is not supported with the AMS Designer simulator.

Note: If you have specified device checks or operating region specifications for a test, ensure that only the Virtuoso Spectre circuit simulator, Virtuoso Accelerated Parallel simulator, or Virtuoso AMS Designer simulator (with Spectre or APS as the solver) is selected for that test. For more information about selecting the simulator for a test, see Choosing the Target Simulator on page 90.

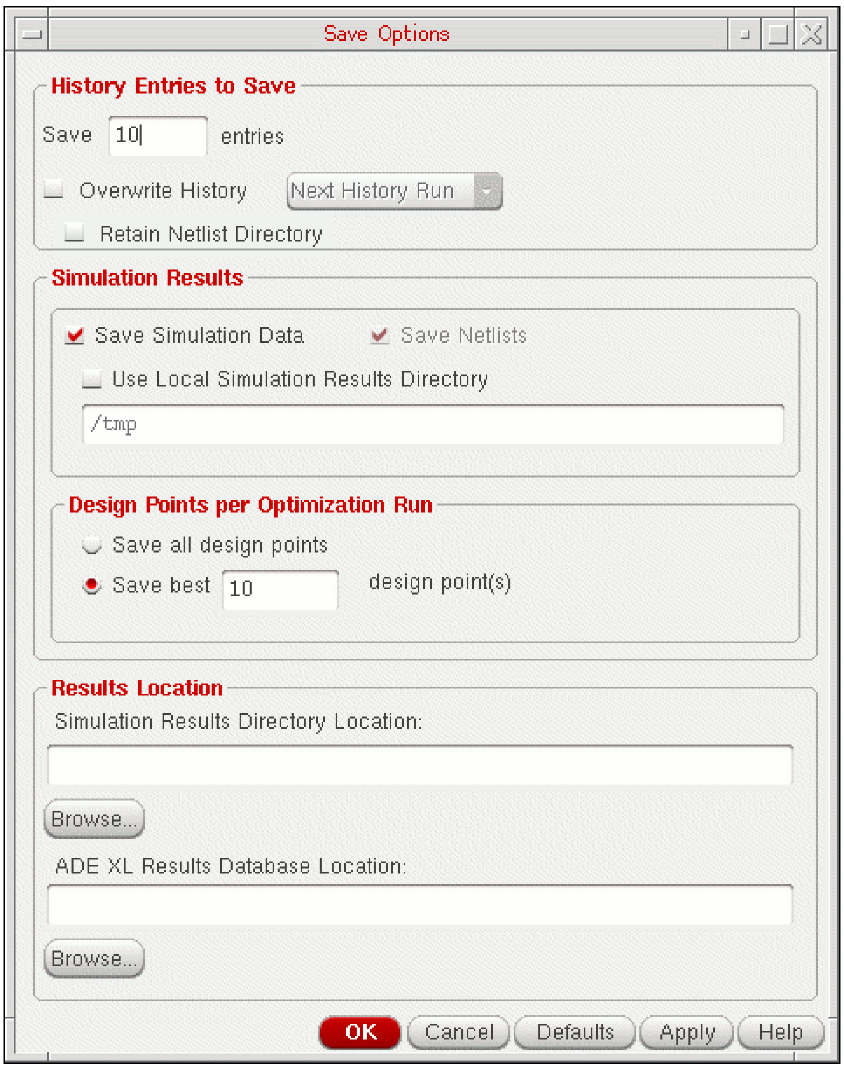
Specifying Options for Saving Simulation Results

By default, the simulation results information is saved to *libraryName*/*cellName*/adexl/results/data/<*history_item*> in the location specified using the asimenv.startup projectDir environment variable. The default setting for this environment variable is $HOME/simulation.

To specify the options for saving simulation results and a different location where you want the program to save the simulation results, do the following:

1. In the ADE XL session window, choose *Options – Save*.

   The Save Options form appears.



   By default, the simulation results and the netlist data are saved for all the design points.

- ❑ If you do not want to save the simulation results data, deselect the *Save Simulation Data* check box.
- ❑ If you do not want to save the simulation results and the netlist data, deselect the *Save Simulation Data* check box, then deselect the *Save Netlists* check box.
- ❑ To save only the simulation results, but not the netlist for each design point, deselect *Save Netlists* and select only the *Save Simulation Data* option. When this option is selected, the point-specific netlist data is not saved and the netlist directory for every point is replaced with a symbolic link to the reference netlist directory. This combination of save options is useful in saving space when the netlist for each design point is very large and space-consuming.

2. By default, the results for distributed simulation runs are saved in the location specified in the *Simulation Results Directory Location* field. To save the results for distributed simulation jobs run on a remote system in a local directory on that system, select the *Use Local Simulation Results Directory* check box and specify the path to the local directory.

Note the following:

- ❑ You must ensure that the specified local directory path exists on all the remote systems on which a distributed simulation is run.
- ❑ If this option is set, you can view the simulation results in the Results tab. However, you cannot plot the results, re-evaluate expressions, or annotate simulation results to the schematic.
- ❑ The results saved on remote systems will not be deleted if you delete the corresponding history items. You must manually delete the directories containing the results on remote systems.

Tip

Setting this option improves distributed simulation performance because the results are saved on the remote systems on which the jobs are run instead of saving them over the network.

3. In the *Simulation Results Directory Location* field, type the directory path where you want the program to write your simulation results information; or do the following:

a. Click *Browse*.

b. On the form that appears, navigate to and select the directory where you want the program to write your simulation results information.

c. Click *Open*.

If your design library is set up as read-only, you can use this field to specify a writable location.

4. Click *OK*.

   The program writes simulation results information to `libraryName/cellName/adexl/results/data/<history_item>` in the specified directory.

   If you do not specify a simulation results location, but specify a ADE XL results database location (see Specifying Results Database Location on page 76), the program writes this information to `libraryName/cellName/adexl/results/data/<history_item>` in the results database location.

Note: You can also specify the simulation results location using the `adexl.results saveResDir` environment variable. For more information, see saveResDir on page 901.

See also

- ADE XL View Directory Structure on page 762
- Specifying Results Database Location on page 76
- Specifying How Much Data to Save on page 805
- Overwriting a History Item during Subsequent Simulation Runs on page 807
- Working with Read-Only ADE XL Views on page 775

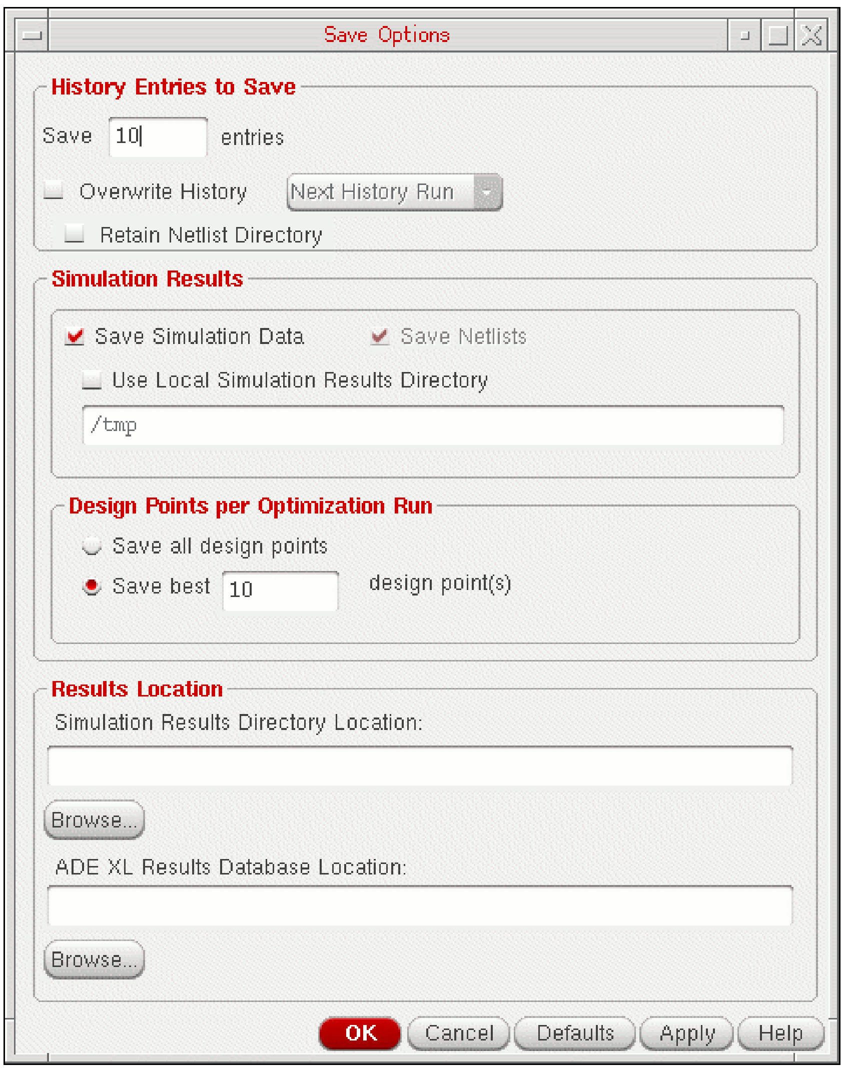
Specifying Results Database Location

By default, the program writes the results database and run log files to *`libraryName`*`/`*`cellName`*`/adexl/results/data` in the ADE XL view. For more information about the ADE XL view, see ADE XL View Directory Structure on page 762.

To specify a different location where you want the program to save the results database and run log files, do the following:

1. In the ADE XL session window, choose *Options – Save*.

   The Save Options form appears.



2. In the *ADE XL Results Database Location* field, type the directory path where you want the program to write your results database information and run log files; or do the following:

   a. Click *Browse*.

   b. On the form that appears, navigate to and select the directory where you want the program to write your results database information and run log files.

   c. Click *Open*.

   If your design library is set up as read-only, you can use this field to specify a writable location.

3. Click *OK*.

   The program writes results database information and run log files to `libraryName/cellName/adexl/results/data` in the specified directory.

   If you do not specify a results database location, and you open the ADE XL view in read-only mode or you do not have write permissions in the ADE XL view, the program writes this information to `libraryName/cellName/adexl/results/data` in the location specified using the `asimenv.startup projectDir` environment variable. The default setting for this environment variable is `$HOME/simulation`. For more information, see Working with Read-Only ADE XL Views on page 775.

Note: You can also specify a different location for the results database and run log files using the `adexl.results saveDir` environment variable. For more information, see saveDir on page 899.

See also

■ ADE XL View Directory Structure on page 762

■ Specifying Options for Saving Simulation Results on page 73

■ Specifying How Much Data to Save on page 805

■ Overwriting a History Item during Subsequent Simulation Runs on page 807

■ Working with Read-Only ADE XL Views on page 775

2

Specifying Tests and Analyses

You can perform the following tasks from the Data tab in the Data View assistant pane in the ADE XL environment:

See also "Data View Assistant Right-Click Menus" on page 126.

Working with Tests

See the following topics:

Adding a Test

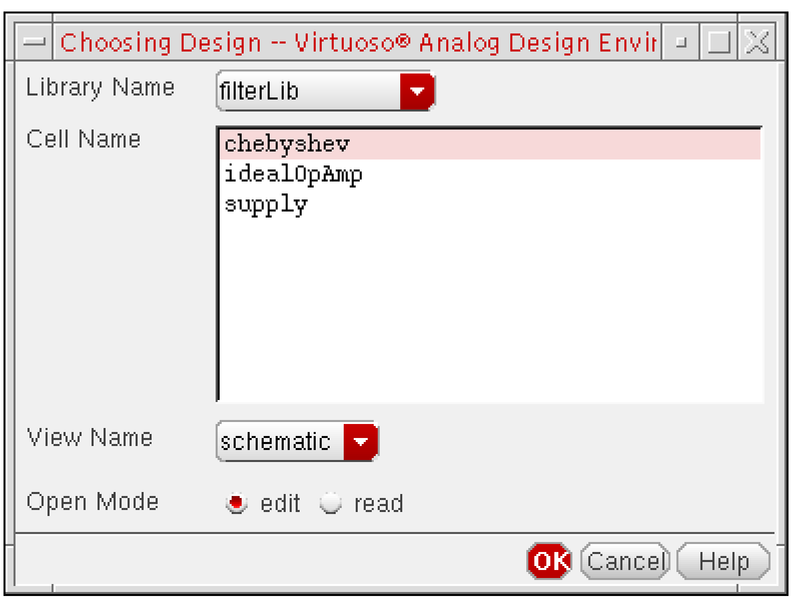
To add a test, do the following:

1. On the Data tab in the Data View assistant pane, right-click a test name and choose *Add Test*.

Tip

Alternatively, you can click where it says *Click to add test* on the Data View assistant pane.

The Choosing Design form appears.



2. In the *Library Name*, *Cell Name*, and *View Name* fields, select a design.

   See "Choosing a Design" on page 89 for more information.

3. Click *OK*.

   Your test appears on the Data View assistant pane.

   The name of the test is *`libName:cellName:sequenceNumber`* where *`sequenceNumber`* starts at `1` and increments by one for every test you add that has the same base name (so that each test name is unique). For example, the first test created in the `ade_sample` cell in the `demoLib` library is named `demoLib:ade_sample:1`, the second test is named `demoLib:ade_sample:2`, and so on.

   The program performs a copy-from-cellview operation so that design variables appear on the Variables and Parameters assistant pane when you first add a test.

You can now define the rest of your test. See "Specifying Tests and Analyses" on page 79 for a list of tasks you can perform on the Data View assistant pane.

Renaming a Test

To rename a test, do the following:

1. Click once to select the test name you want to change.
2. Click again to make the name editable.

   The entire test name is highlighted.

3. Either type a completely new name for the test or click to place the edit cursor and edit the name.
4. Press *Return*.

   The new test name appears on the Data View assistant pane.

Copying a Test

To reuse the settings of an existing test, you can create a copy of the test. This allows you to quickly setup new tests.

To copy a test, do the following:

➤ On the Data View assistant pane, right-click the test you want to copy and choose *Create Test Copy*.

The new test name appears on the Data View assistant pane.

The name of the new test has the suffix *`:sequenceNumber`* where *`sequenceNumber`* starts at `1` and increments by one for every copy of the original test (so that the test name of each copy is unique). For example, if the original test has the name `demoLib:ade_sample:1`, the first copy of the test is named `demoLib:ade_sample:1:1`, the second copy of the original test is named `demoLib:ade_sample:1:2` and so on.

Removing a Test

Caution

There is no undo for this action.

To remove a test, do the following:

➤ On the Data View assistant pane, right-click the test you want to remove and choose *Delete*.

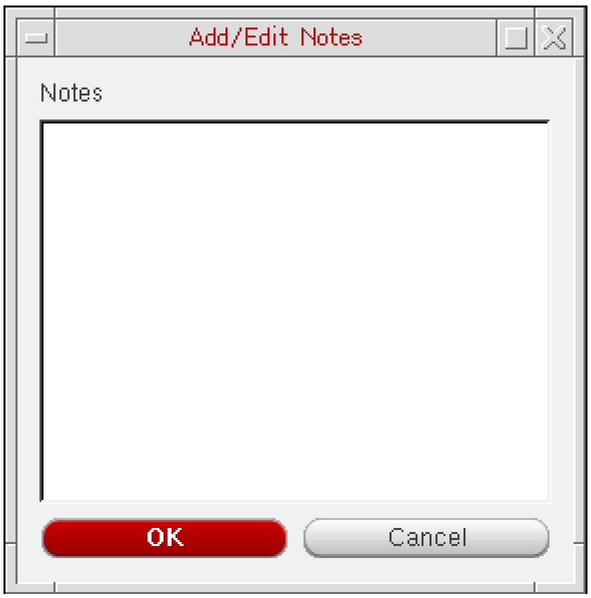
The program deletes the selected test.

Adding Notes to a Test

To add notes to a test, do the following:

1. On the Data View assistant pane, right-click the test for which you want to add notes and choose *Notes*.

   The Add/Edit Notes form is displayed.



2. In the *Notes* field, add notes for the test.

   Note: By default, the notes field can accept only 512 characters. You can change this limit by setting the maxNotesLength environment variable.

3. Click *OK*.

   The notes are added to the test details in the ADE XL setup database. Similarly, you can add notes for global variables, parameters, corners, Reliability Analysis setup, or history items.

Important Points to Remember

- Notes added to a test are visible in the tooltip information, which is displayed when you hover the pointer over a test name on the Data View assistant pane. If the notes are very long, only the first 10 rows are displayed in the tooltip by default. You can change this limit by setting the maxNotesRowsDisplay environment variable.
- To remove notes added to a test, right-click on the test name and choose *Delete Notes*.

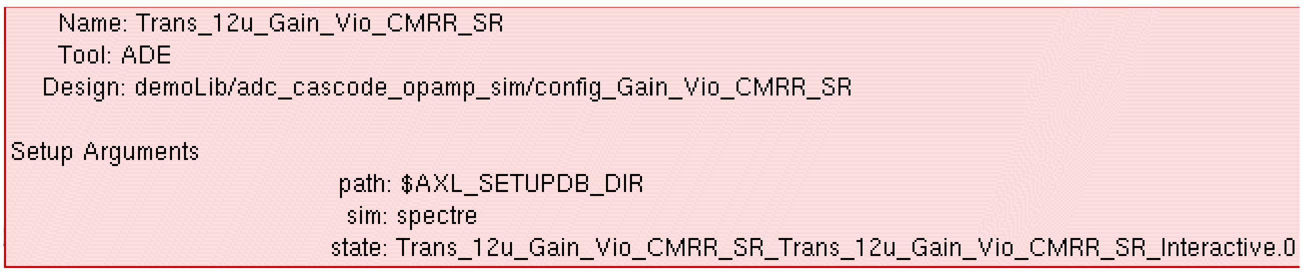
Viewing Information about a Test

You can view information about a test in ADE XL such as the test name, the design associated with the test (lib/cell/view), the target simulator, and the ADE state.

To view information about a test, do the following:

➤ On the Data View assistant pane or on the Variables tab of the Variables and Parameters assistant pane, hover the mouse cursor over a test name.

A text box containing test information appears.



If you have added notes for the test, they are appended to the test information.

After a while, or if you move the mouse cursor, the text box disappears.

Adding, Changing, and Removing Analyses

See the following topics:

- Adding an Analysis on page 85
- Changing an Analysis on page 87
- Removing an Analysis on page 87

Adding an Analysis

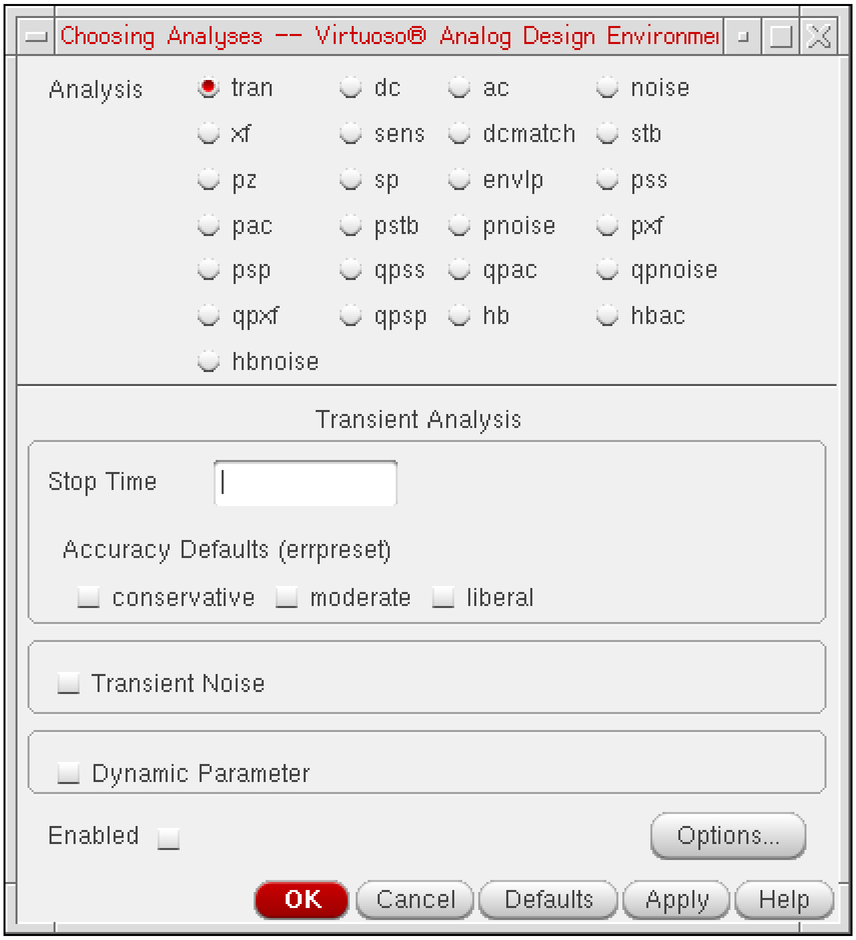
To add an analysis to your test, do the following:

1. On the Data View assistant pane, right-click the test and choose *Add Analysis*.

Tip

Alternatively, you can click where it says *Click to add analysis* in an expanded test tree on the Data View assistant pane.

The Choosing Analyses form appears.



The items that appear on this form depend on the simulator you selected (see "Choosing the Target Simulator" on page 90).

2. In the *Analysis* section, select a radio button for the analysis type you want to define.

   The setup parameters corresponding to the analysis you selected appear on the form.

3. Set the analysis parameters you want (see your simulator's user guide for more information).

   Your analysis is enabled by default: After you specify one analysis parameter, the *Enabled* check box is selected.

4. Click *OK*.

   The analysis you specified appears under the test name on the Data View assistant pane.

Changing an Analysis

To change an analysis setup, do the following:

1. On the Data View assistant pane, double-click the analysis you want to change.

   Note: You might need to expand the tree by clicking the plus sign (+) to the left of the test that contains the analysis you want to change.

   The Choosing Analyses form appears.

   The items that appear on this form depend on the simulator you selected (see "Choosing the Target Simulator" on page 90).

Tip

Alternatively, you can change an analysis by right-clicking its name and choosing *Edit* from the pop-up menu that appears.

2. (Optional) In the *Analysis* section, select a radio button to change the analysis type you want to define.

   The setup parameters corresponding to the analysis you selected appear on the form.

3. (Optional) Change the analysis parameters you want (see your simulator's user guide for more information).

4. (Optional) You can deselect the *Enabled* check box to turn off the analysis without removing the setup information.

Tip

Alternatively, you can turn off an analysis by deselecting the check box next to the analysis name in the tree on the Data View assistant pane.

5. Click *OK*.

Removing an Analysis

Caution

There is no undo for this action.

To remove a analysis, do the following on the Data View assistant pane:

1. To the left of the test containing the analysis you want to remove, click *+* to expand the test (if it is not already expanded).
2. Right-click the analysis you want to remove and choose *Delete*.

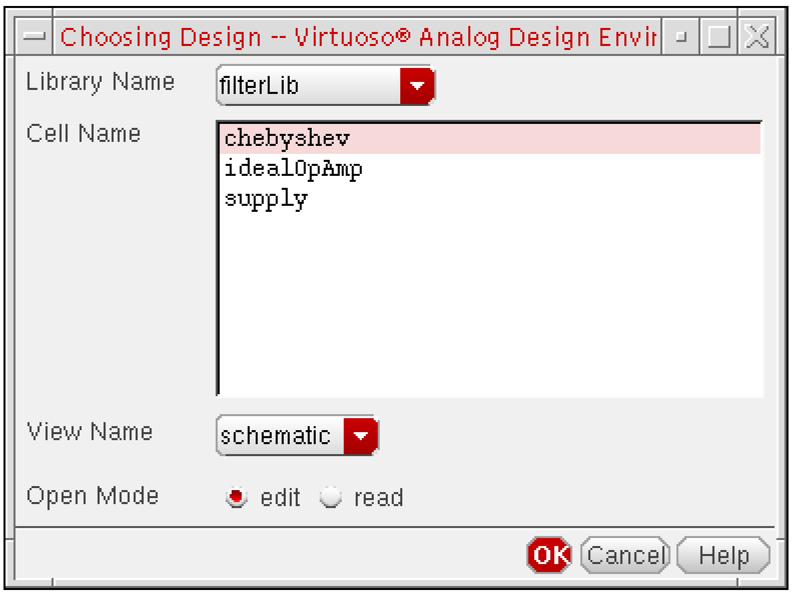
   The program removes the selected analysis.

Choosing a Design

To specify a design for a test, do the following:

1. On the Data View assistant pane, right-click the test or analysis name and choose *Design*.

   The Choosing Design form appears.



2. In the *Library Name* drop-down list, select a library name.
3. In the *Cell Name* list box, select a cell name.
4. In the *View Name* drop-down list, select a view name.
5. For the *Open Mode*, select one of the following radio buttons:
   - *edit* opens the design in edit mode.
   - *read* opens the design in read-only mode
6. Click *OK*.

Opening a Design Schematic

To open the design associated with a test, do the following:

➤ On the Data View assistant pane, right-click the test and choose *Open Design in Tab*.

The design associated with the test appears on a new tab in your session window.

Tip

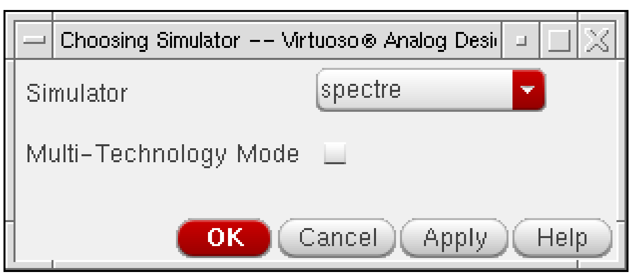
When the design is open, press `Ctrl+Tab` to switch between the ADE XL and the schematic view.

Choosing the Target Simulator

To choose the target simulator for a test, do the following:

1. On the Data View assistant pane, right-click the test or analysis name and choose *Simulator*.

   The Choosing Simulator form appears.



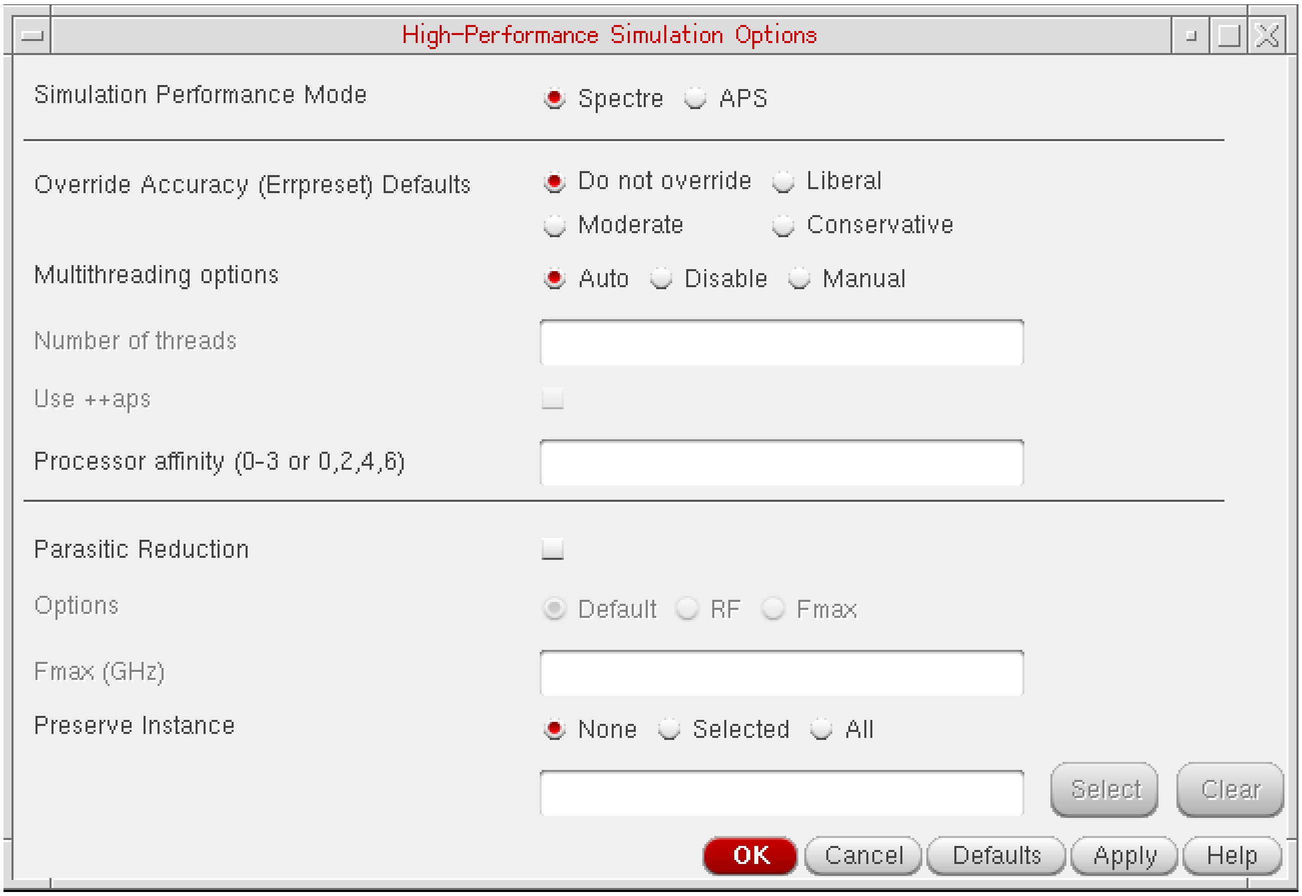
   Note: For information about *Multi-Technology Mode*, see "Enabling Multi-Technology Simulation" in the *Virtuoso Analog Design Environment GXL User Guide*.

2. In the *Simulator* drop-down list, select a simulator.
3. Click *OK*.

To specify options for high performance simulation, do the following:

1. On the Data View assistant pane, right-click the test or analysis name and choose *High-Performance Simulation*.

The High-Performance Simulation Options form appears.



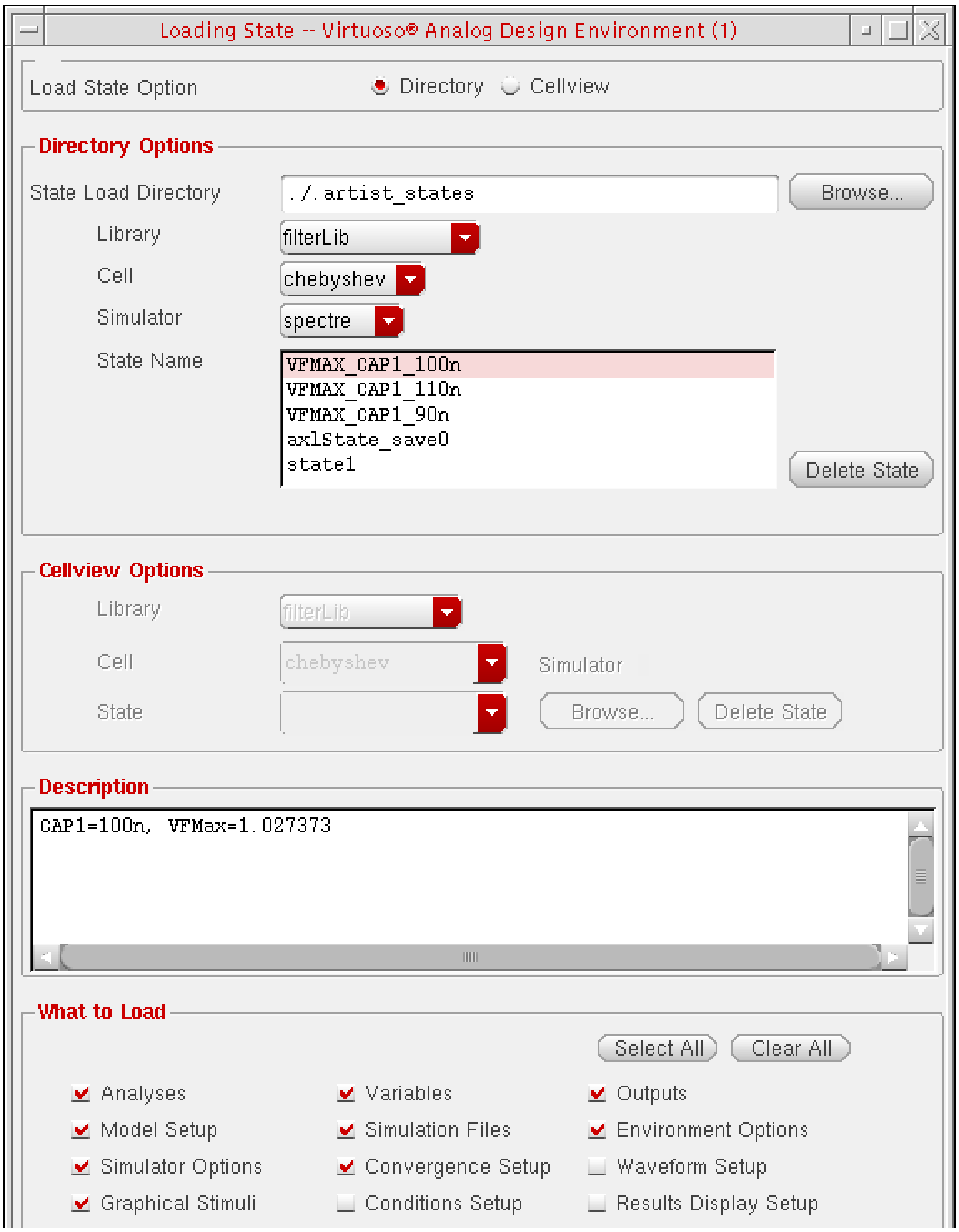
For more details on the various options on this form, refer to Specifying Performance and Parasitic Reduction Options in the *Virtuoso Analog Design Environment L User Guide*.

Loading State Information

To load state information to a test, do the following:

1. On the Data View assistant pane, right-click the test or analysis name and choose *Load State*.

   The Loading State form appears.



In the *Load State Option* group box, the *Directory* radio button is selected by default.

2. In the *Directory Options* group box, select a state load directory by choosing information from the various drop-down lists or by clicking *Browse* to open a directory browser window and navigating to your state directory.

   State names appear in the *State Name* list box.

3. In the *State Name* list box, select the name of the state you want to load.
4. In the *What to Load* group box, select the check box for each type of information you want to load. For example, if you want to load analyses from your state file, select the *Analyses* check box; if you want to load design variables, select the *Variables* check box.

Tip

You can click *Select All* to select all check boxes or *Clear All* to clear them.

5. Click *OK*.

   The state information you selected appears on the appropriate assistant panes in your environment according to the following guidelines:

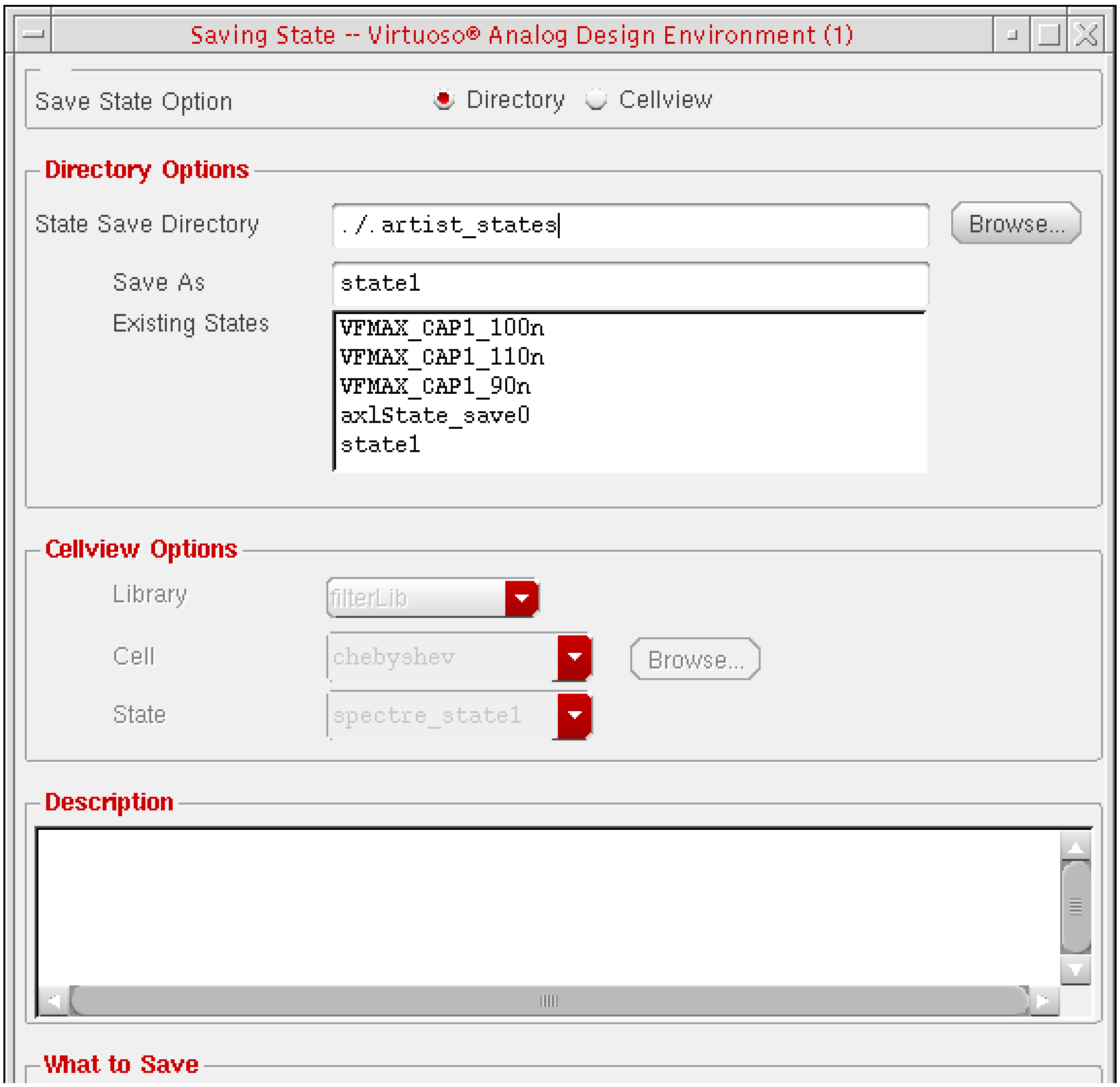
   - The program ignores state information that is incompatible with the selected simulator.
   - The program overwrites state variables with matching name and value types.
   - An error message appears if the state variable name is the same but the value type is different.

Saving State Information

To save state information, do the following:

1. On the Data View assistant pane, right-click the test or analysis name and choose *Save State*.

   The Saving State form appears.



   State names appear in the *State Name* list box.

2. In the *Save As* field, type a name for the state you want to save.

3. In the *What to Save* group box, select the check box for each type of information you want to save. For example, if you want to save analyses to your state file, select the *Analyses* check box; if you want to save design variables, select the *Variables* check box.

Tip

You can click *Select All* to select all check boxes or *Clear All* to clear them.

Note: If you save the analyses, you can load your saved state only for tests that have the same target simulator. If you do not save the analyses, you can restore your state for a test that has a different target simulator.

4. Click *OK*.

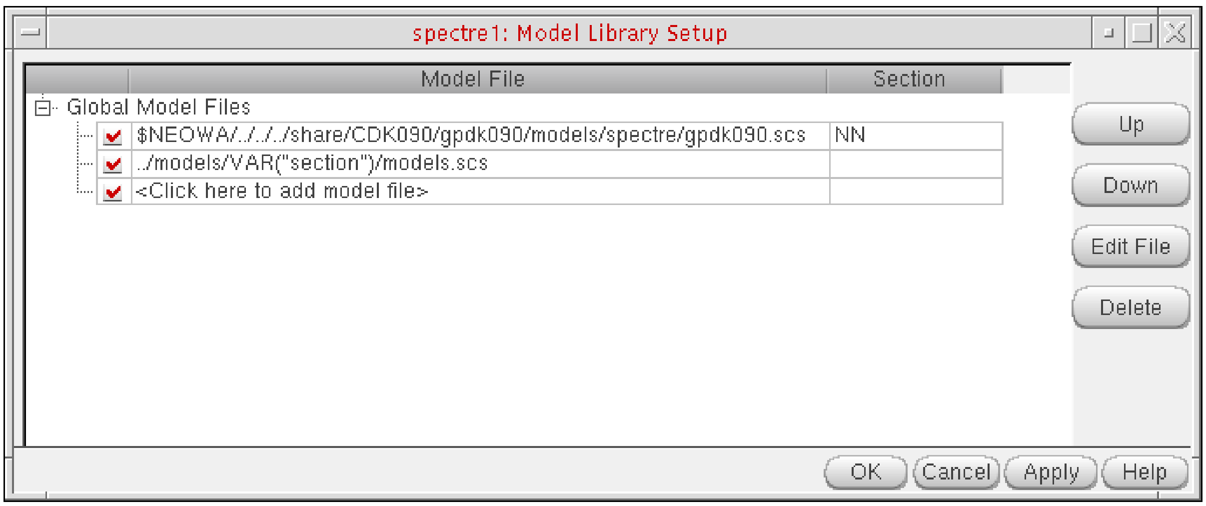
   The program saves the state information you selected to the state name you specified.

Specifying Model Libraries

To specify the model libraries to use when simulating a test, do the following:

1. On the Data View assistant pane, right-click the test or analysis name and choose *Model Libraries*.

   The Model Library Setup form appears.



2. In the *Model File* column, type the path and file name of the model file you want to use.

   Alternatively, you can click *...* to open the Choose Model File form where you can navigate to select a valid model file.

   See Chapter 9, "Working with Model Files and CDF" for more information about model files.

3. (Optional) In the *Section* column, select a section from the drop-down list.

   A model file can have zero or more sections. If a model file contains no sections, there is no drop-down list available. The section you select determines which model definition the simulator uses.

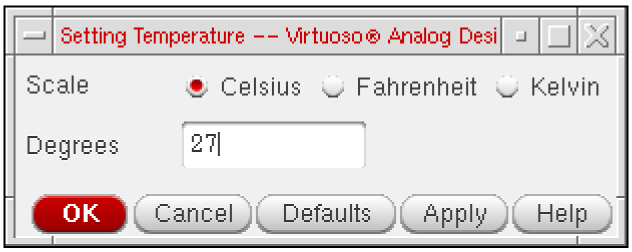
   For more information about sections in model files, see "Corners Modeling" in the *Direct Simulation Modeling User Guide*.

Setting the Simulation Temperature

To set the simulation temperature for a test, do the following:

1. On the Data View assistant pane, right-click the test or analysis name and choose *Temperature*.

   The Setting Temperature form appears.

2. Select one of the following *Scale* radio buttons to indicate the units you want to use for temperature:
   - *Celsius*
   - *Fahrenheit*
   - *Kelvin*
3. In the *Degrees* field, type a value.

   The value you type can be a constant or expression which can include the use of `VAR`.
4. Click *OK*.

Specifying Analog Stimuli

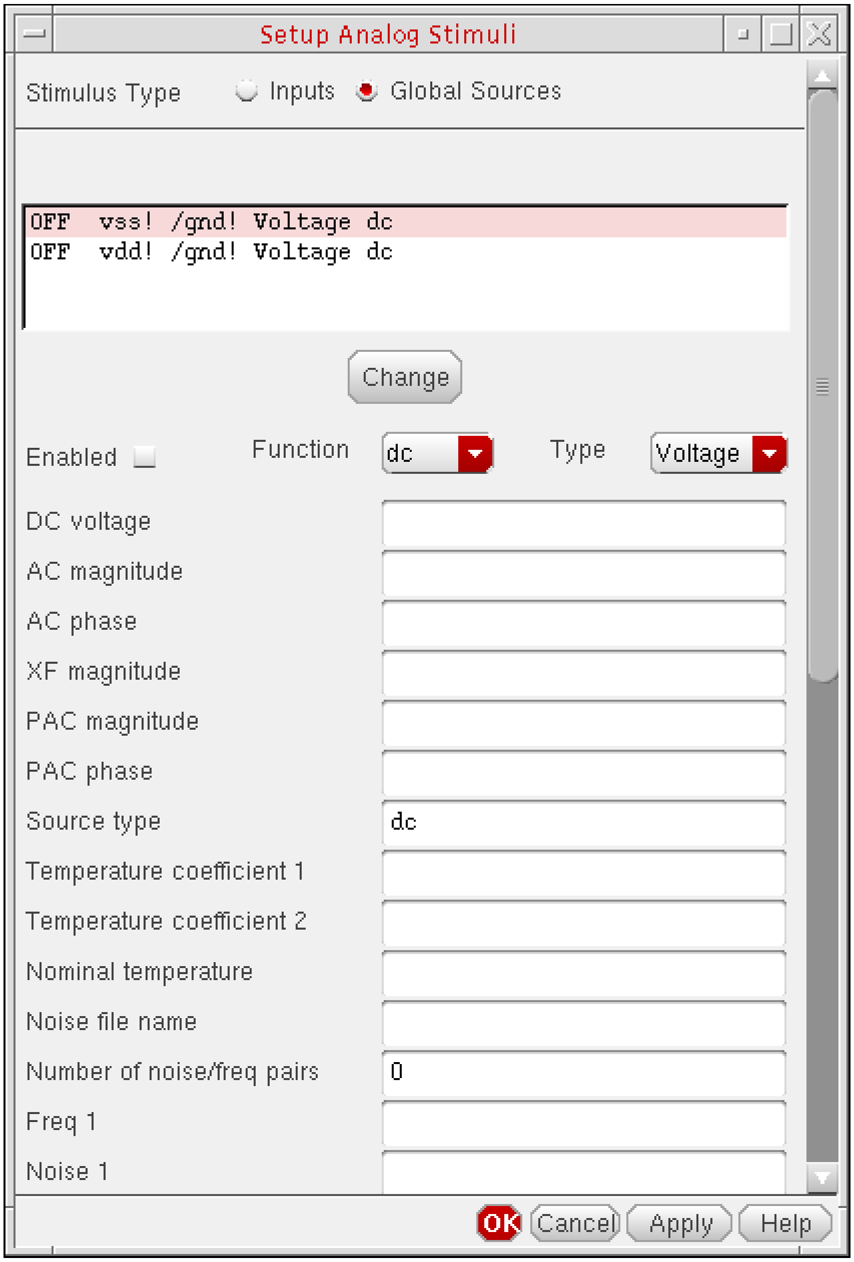
To specify analog stimuli (input stimulus and global sources) for a test, do the following:

Important

The top-level schematic of your design must have input pins, bidirectional pins, or global nets (such as `vdd!` for power stimulus).

1. On the Data View assistant pane, right-click the test or analysis name and choose *Stimuli*.

The Setup Analog Stimuli form appears.



Current values for the selected input, global source, or bidirectional pin appear in the fields on the form. The fields that appear depend on what is required for your simulator. When the form first appears, these fields might be blank, might contain default values, or might contain initial values that you specified at another time.

2. Select one of the following *Stimulus Type* radio buttons:

   - ❑ *Inputs* indicates that you want to apply stimulus to input or bidirectional pins

     All input and bidirectional pins in your design appear in the list box.

   - ❑ *Global Sources* indicates that you want to specify global power sources

     All global sources on your schematic (excluding the `gnd!` ground signal) appear in the list box. You can set DC source values here.

3. In the list box, select the pin or global source whose parameters you want to specify.

4. In the *Function* drop-down list, select one of the following source functions:

Important

If you selected a global source, you must select *dc*.

dc	Direct current
sin	Sinusoidal waveform
pulse	Pulse waveform
exp	Exponential waveform
pwl	Piecewise linear waveform
pwlf	Piecewise linear waveform file
sffm	Single frequency FM source waveform

5. In the *Type* field, specify whether your stimulus is a *Voltage* or *Current* stimulus.

6. In the remaining fields on the form, type the parameter values you want.

   The parameters displayed in this list depend on the simulator you are using. See your simulator documentation for details on setting these parameters.

7. Click *Change*.

   The list box displays the signal using the proper stimulus syntax for your simulator.

8. Repeat the above steps for each pin to which you want to apply external stimulus.

9. Click *OK*.

The simulation environment recognizes your external stimuli, creates a simulator input file, and generates a stimulus file containing input and power source stimuli in the proper syntax for your simulator.

Tip

You can remove the voltage source for a global signal by deselecting the *Enabled* check box. When you click *Apply*, the status that appears in the list box (as the leftmost token) changes from `ON` to `OFF`. The netlister continues to honor the signal's presence and connectivity in your design.

See also

- Specifying a Custom Library of Sources on page 102
- Specifying Simulation Files on page 103

Specifying a Custom Library of Sources

Unless you specify otherwise, all sources, whether used for stimulus or for a power supply, come from `analogLib`. To specify a custom library of sources, do the following:

1. In the CIW, choose *Tools – Library Manager*.

   The Library Manager form appears.

2. From the *Library* list box, select the library of the current design.

3. Choose *Edit – Properties*.

   The Library Property Editor form appears.

4. If the *refLibs* property does not already exist, create it as follows:

   a. On the Library Property Editor form, click *Add*.

      The Add Property form appears.

   b. In the *Name* field, type `refLibs`.

   c. In the *Type* drop-down list, select *string*.

      The *Value* and *Possible Choices* fields appear.

   d. Click *OK*.

      The *refLibs* property appears on the Library Property Editor form.

5. In the value field for *refLibs*, type a list of one or more libraries in the search order you want.

6. Click *OK*.

7. On the Library Manager form, choose *File – Exit*.

Specifying Simulation Files

Important

You must specify any simulation files you want to use prior to running a simulation.

To specify simulation files for a test, do the following:

1. On the Data View assistant pane, right-click the test or analysis name and choose *Simulation Files*.

   The Simulation Files Setup form appears.

   You can set up include paths, definition files and stimulus files using the Paths/Files tab and setup digital vector files using the Vector Files tab.

   Note: The fields that appear on the form depend on your target simulator.

Figure 2-1 Paths/Files Tab

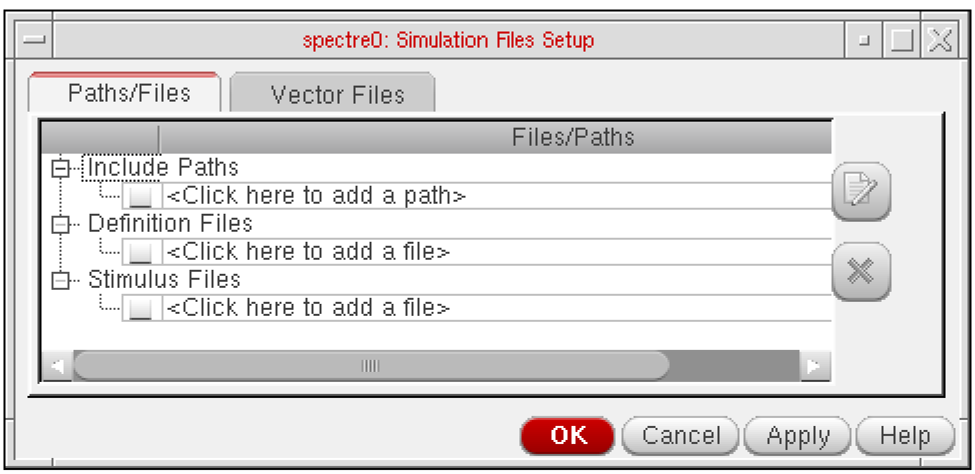
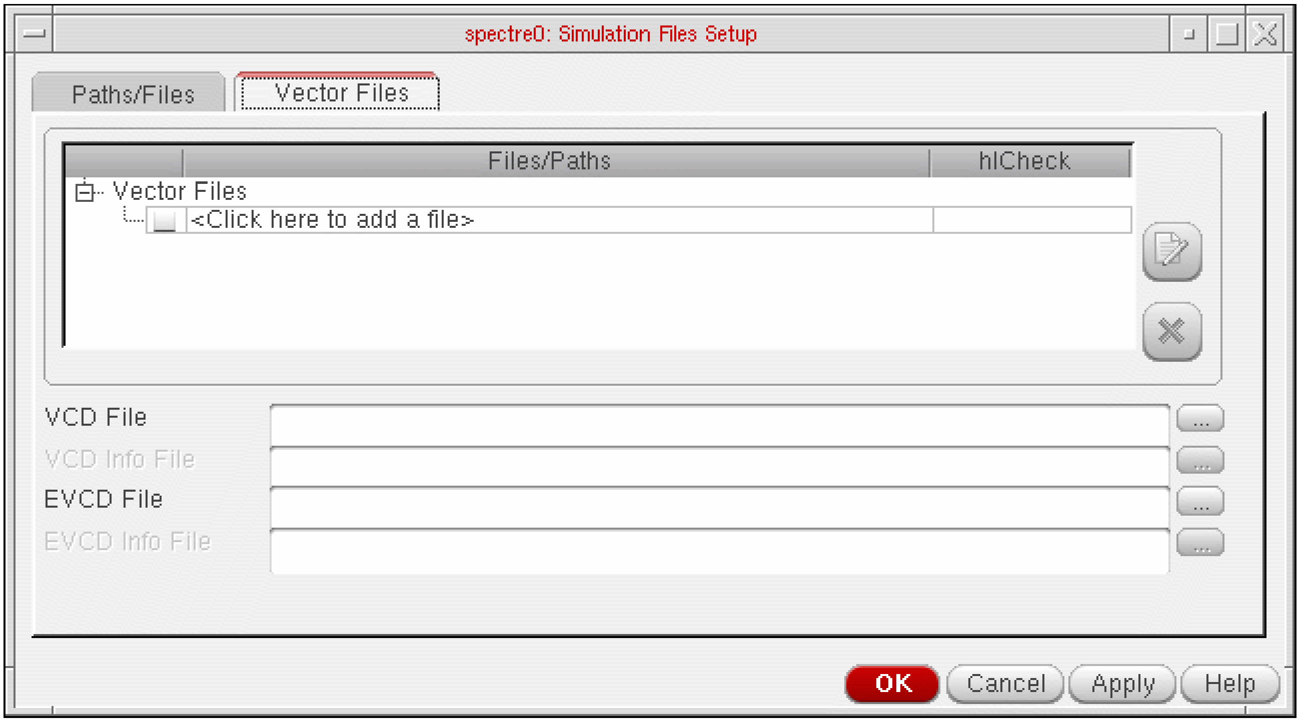


Figure 2-2 Vector Files Tab



See the following sections for more information about using the Simulation Files Setup form:

- Setting Up Include Paths on page 104
- Setting Up Definition Files on page 105
- Setting Up Stimulus Files on page 105
- Setting Up Vector Files on page 106
- Setting Up VCD and EVCD Files on page 106
- Enabling and Disabling Simulation Files on page 107
- Editing Simulation Files on page 107
- Deleting Simulation Files on page 107

Setting Up Include Paths

Include paths specify the directories that contain the files you want to include when simulating your design.

To set up include paths, do the following:

1. In the Simulation Files Setup form, click the Paths/Files tab.
2. In the *Include Paths* tree, click where it says *<Click here to add a path>* and type the path to the directory, or click the browse button to select the directory using the Choose Files/Paths form.

   The simulator resolves a relative path by looking in the `netlist` directory (relative to where you run the simulation) first. If the path starts with the `.` character, the simulator also resolves this by looking in the `netlist` directory first, then in each of the directories specified in the *Include Path* in the order you type them. The `.` does not mean the current directory.

Setting Up Definition Files

Definition files contain function and parameter definitions that are not displayed in the *Design Variables* section of the simulation window. See the following sample file that contains function and parameter definitions for the Spectre circuit simulator.

`<your_install_dir>/tools/dfII/samples/artist/models/spectre/definitions.scs`

The parameters in this file are referenced by included models and are not referenced from any part of the design in the Cadence library.

To set up definition files, do the following:

1. In the Simulation Files Setup form, click the Paths/Files tab.
2. In the *Definition Files* tree, click where it says *<Click here to add a file>* and type the path and file name of your definition file, or click the browse button to select one or more files using the Choose Files/Paths form.

Setting Up Stimulus Files

Stimulus files can contain input and power supply stimuli, initialize nodes, and include estimated parasitics in the netlist. You can look at the following example file that contains Spectre circuit simulator stimuli definitions for the opamp sample design in the aExample library.

`<your_install_dir>/tools/dfII/samples/artist/models/spectre/opampStimuli.scs`

In your stimulus file, you can type node names and component names using Open Simulation System (OSS) syntax `[#name]` and the system will substitute the corresponding node numbers when writing the netlist. You can use a backslash (\) to escape a square bracket. For information about OSS syntax, see the *Open Simulation System Reference*.

To set up stimulus files, do the following:

1. In the Simulation Files Setup form, click the Paths/Files tab.
2. In the *Stimulus Files* tree, click where it says *<Click here to add a file>* and type the path and file name of your stimulus file, or click the browse button to select one or more files using the Choose Files/Paths form.

Setting Up Vector Files

To set up vector files, do the following:

1. In the Simulation Files Setup form, click the Vector Files tab.
2. In the *Vector Files* tree, click where it says *<Click here to add a file>* and type the path and file name of your digital vector file, or click the browse button to select one or more files using the Choose Files/Paths form.
3. Click on the *hlCheck* field next to the file and do one of the following:
   - ❑ Choose 1 to enable the check for H and L states for input signals (`HLCheck`).
   - ❑ Choose 0 to disable the check for H and L states for input signals (`HLCheck`)

Setting Up VCD and EVCD Files

For information about VCD and EVCD stimuli, see "Verilog Value Change Dump Stimuli" (Chapter 12) in the *Virtuoso UltraSim Simulator User Guide*.

To setup VCD and EVCD files, do the following:

1. In the Simulation Files Setup form, click the Vector Files tab.
2. In the *VCD File* field type the path and file name of your Verilog value change dump (VCD) file name, or click the browse button to select the file using the Choose a File form.

   The *VCD Info File* field becomes active.
3. In the *VCD Info File* field type the path and file name of your signal information file, or click the browse button to select the file using the Choose a File form.
4. In the *EVCD File* field type the path and file name of your "Extended" VCD (EVCD) file, or click the browse button to select the file using the Choose a File form.
5. The *EVCD Info File* field becomes active.

6. In the *EVCD Info File* field type the path and file name of your signal information file, or click the browse button to select the file using the Choose a File form.

Enabling and Disabling Simulation Files

➤ To enable a simulation file for simulation, select the check box next to it.

Note: By default, the simulation files you add are enabled for simulation.

➤ To disable a simulation file for simulation, clear the check box next to it.

Editing Simulation Files

To edit simulation files, do the following:

1. Select one or more simulation files.
2. Click the  button.

   The simulation files are opened in a text editor.

Deleting Simulation Files

To delete simulation files, do the following:

1. Select one or more simulation files.
2. Click the  button.

Specifying Simulation Environment Options

To specify environment options for a test, do the following:

1. On the Data View assistant pane, right-click the test or analysis name and choose *Environment*.

   The Environment Options form appears.

   The fields that appear on the form depend on your target simulator as follows:

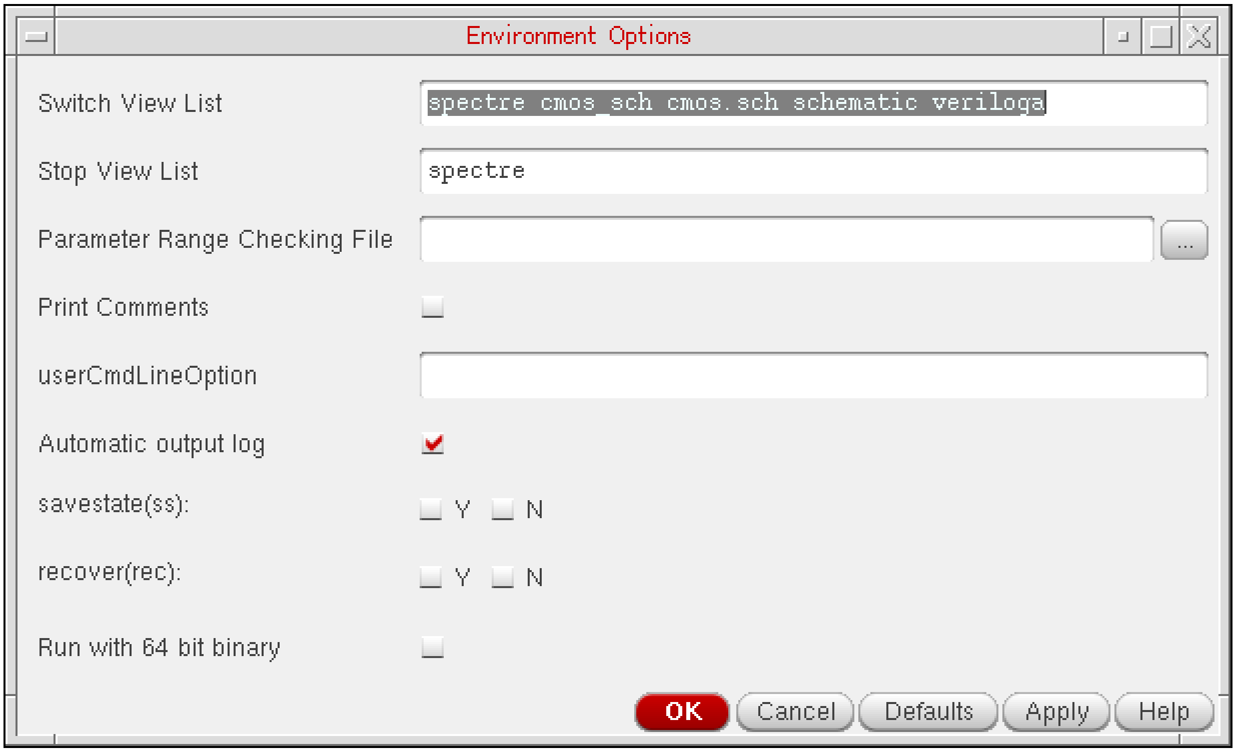
   - ❑ Environment Options for the Spectre Circuit Simulator on page 109
   - ❑ Environment Options for the UltraSim Circuit Simulator on page 110
   - ❑ Environment Options for the AMS Designer Simulator on page 111
   - ❑ Environment Options for the SpectreVerilog Circuit Simulator on page 113
   - ❑ Environment Options for the UltraSimVerilog Circuit Simulator on page 114
   - ❑ Environment Options for the hspiceD Circuit Simulator on page 115

2. When you are done specifying options, click *OK*.

   The environment options you specified are applied to future simulations.

Environment Options for the Spectre Circuit Simulator

You can specify the following options for the Spectre circuit simulator:



- *Switch View List*
- *Stop View List*
- *Parameter Range Checking File*
- *Print Comments*
- *userCmdLineOption*
- *Automatic output log*
- *savestate (ss)*
- *recover (rec)*

Return to main procedure.

Environment Options for the UltraSim Circuit Simulator

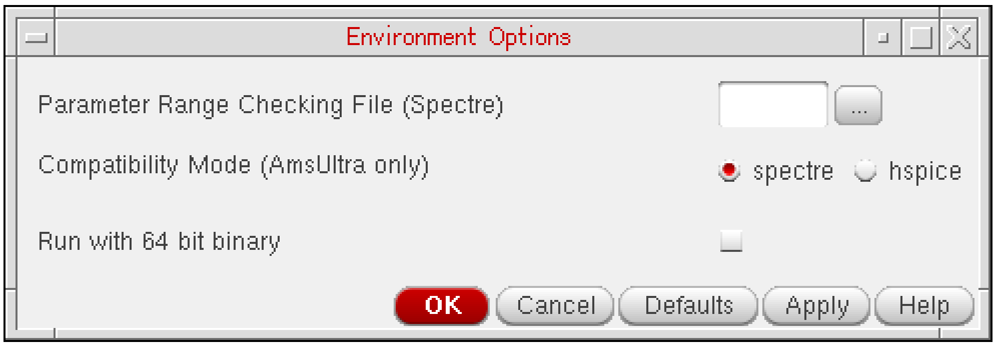
You can specify the following options for the UltraSim circuit simulator:

- *Switch View List*
- *Stop View List*
- *Print Comments*
- *userCmdLineOption*
- *Automatic output log*
- *Netlist Format*
- *Run Mode*

Return to main procedure.

Environment Options for the AMS Designer Simulator

You can specify the following options for the AMS Designer simulator:

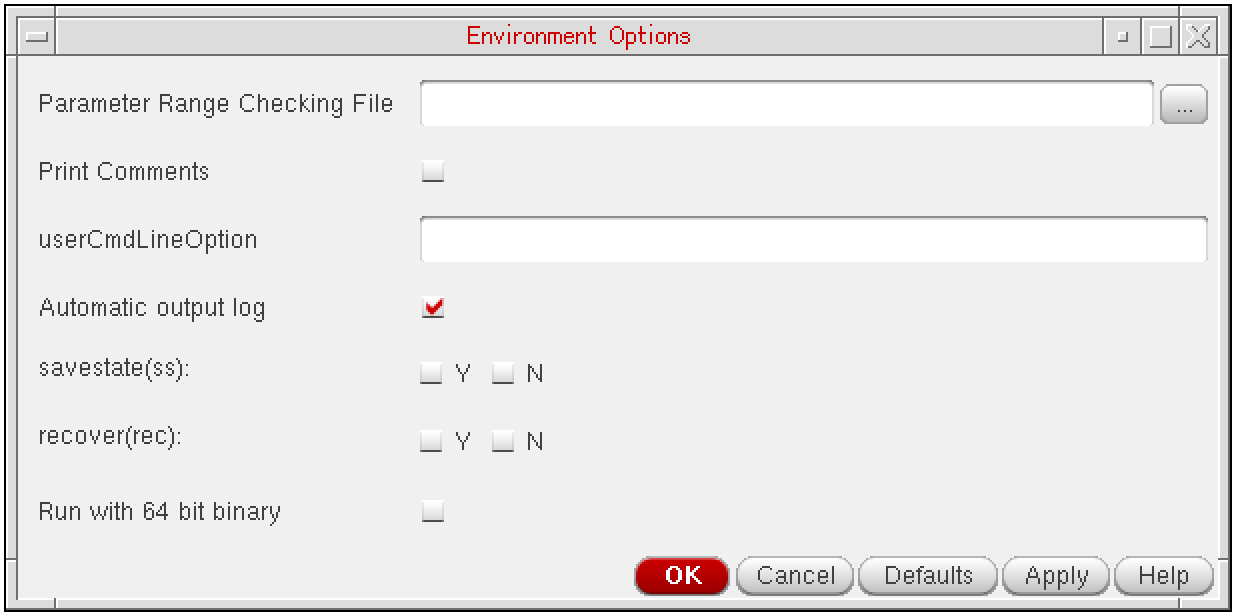


- *Parameter Range Checking File*

Return to main procedure.

Environment Options for the Accelerated Parallel Simulator

You can specify the following options for the Accelerated Parallel Simulator:



- *Parameter Range Checking File*
- *Print Comments*
- *userCmdLineOption*
- *Automatic output log*
- *savestate (ss)*
- *recover (rec)*

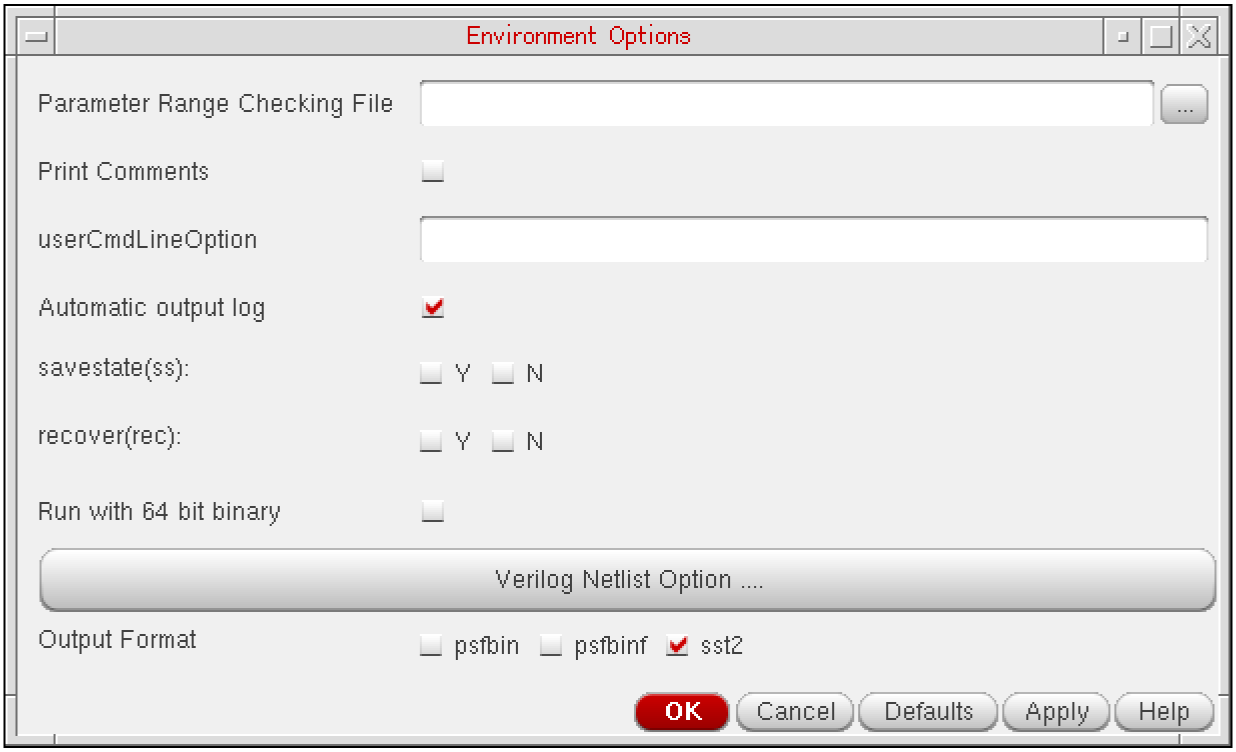
Return to main procedure.

Environment Options for the SpectreVerilog Circuit Simulator

Important

The SpectreVerilog circuit simulator is available only in IC6.1.6 release.

You can specify the following options for the SpectreVerilog circuit simulator:



- *Parameter Range Checking File*
- *Print Comments*
- *userCmdLineOption*
- *Automatic output log*
- *savestate (ss)*
- *recover (rec)*
- *Verilog Netlist Option*

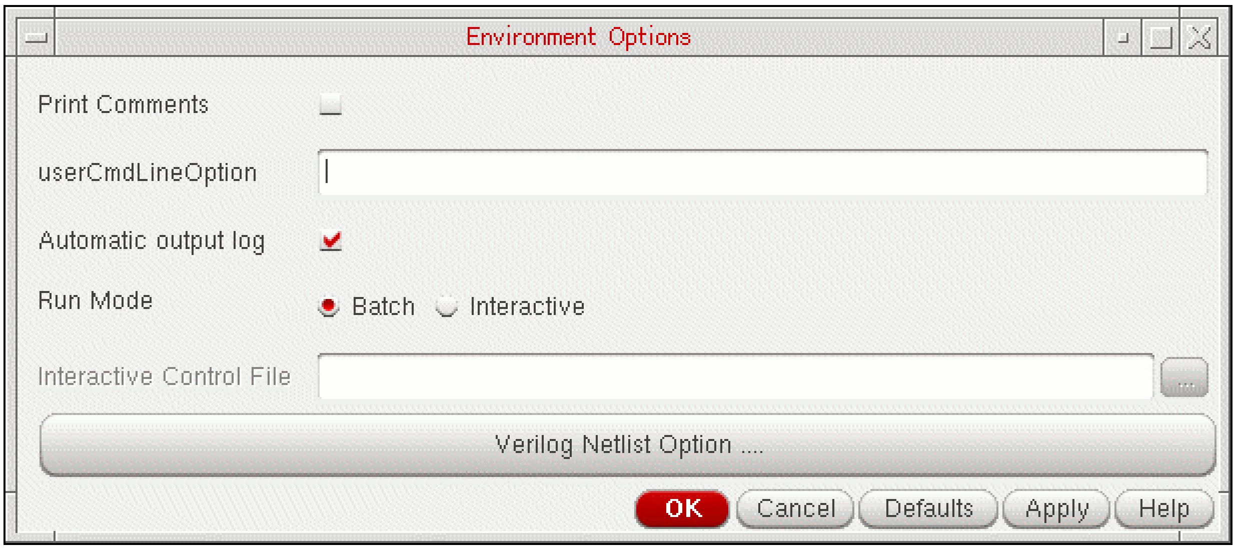
Return to main procedure.

Environment Options for the UltraSimVerilog Circuit Simulator

Important

The UltrasimVerilog circuit simulator is available only in IC6.1.6 release.

You can specify the following options for the UltraSimVerilog circuit simulator:

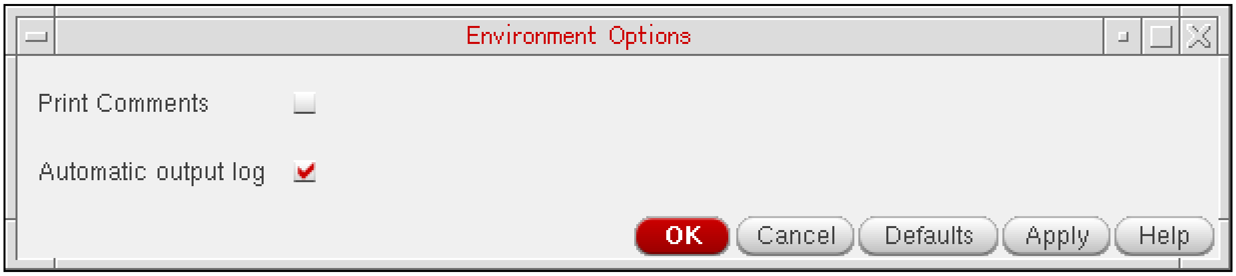


- *Print Comments*
- *userCmdLineOption*
- *Automatic output log*
- *Run Mode*
- *Verilog Netlist Option*

Return to main procedure.

Environment Options for the hspiceD Circuit Simulator

You can specify the following options for the hspiceD circuit simulator:



- *Print Comments*
- *Automatic output log*

Return to main procedure.

Switch View List

The *Switch View List* field appears on the Environment Options form for Spectre, UltraSim, and hspiceD circuit simulators. The switch view list tells the netlister the sequence to use when descending into different views of your design. The flowchart in the Netlisting chapter shows how the netlister selects views when expanding a design. The software searches through hierarchical views in the order you specify. See also "Expanding Hierarchy to Netlist a Design" on page 365.

➤ (Optional) In the *Switch View List* field, type a list of one or more view names to search when looking for design variables and when netlisting.

Stop View List

The *Stop View List* field appears on the Environment Options form for Spectre, UltraSim, and hspiceD circuit simulators.

➤ (Optional) In the *Stop View List* field, type a list of one or more view names (in no particular order) that you want the netlister to use as stopping views.

Print Comments

The *Print Comments* check box appears on the Environment Options form for Spectre, UltraSim, SpectreVerilog, UltraSimVerilog, and hspiceD circuit simulators.

➤ (Optional) Select the *Print Comments* check box if you want the program to write extra comments regarding component location and name to your netlist.

Automatic Output Log

The *Automatic output log* check box appears on the Environment Options form for Spectre, UltraSim, SpectreVerilog, UltraSimVerilog, and hspiceD circuit simulators.

➤ (Optional) Select the *Automatic output log* check box if you want the output log to appear automatically when the simulator generates messages.

Parameter Range Checking File

The *Parameter Range Checking File* field appears on the Environment Options form for Spectre, SpectreVerilog, and AMS circuit simulators.

➤ (Optional) In the *Parameter Range Checking File* field, type the name of the file containing the parameter range limits.

You do not need to type the full path for the file if it is in the directory specified in the *Include Path* field on the Simulation Files Setup form. The program interprets the `.` character in a UNIX path specification relative to the directory where you started the environment.

userCmdLineOption

The *userCmdLineOption* field appears on the Environment Options form for Spectre, UltraSim, SpectreVerilog, and UltraSimVerilog circuit simulators.

➤ (Optional) In the *userCmdLineOption* field, type any command-line options you want to send to the simulator.

savestate (ss)

The *savestate (ss)* check boxes appear on the Environment Options form for Spectre and SpectreVerilog circuit simulators.

➤ Select one of the following *savestate (ss)* check boxes:

- ❑ *Y* indicates that you want to run spectre with the `+ss` option, which causes the simulator to save the states to a file when receiving interrupt signals like QUIT, TERM, INT, or HUP.

- *N* indicates that you want to run spectre with the `-ss` option, which disables the savestate option.

Start from Checkpoint File (rec)

The *Start from Checkpoint File (rec)* check boxes appear on the Environment Options form for Spectre and SpectreVerilog circuit simulators.

➤ Select one of the following *Start from Checkpoint File (rec)* check boxes:

- *Y* indicates that you want to run spectre with the `+rec` option, which restarts a transient simulation based on conditions specified in a checkpoint file.
- *N* indicates that you want to run spectre with the `-rec` option, which does not restart a transient simulation, even if conditions for this have been specified in a checkpoint file.

Netlist Format

The *Netlist Format* radio buttons appear on the Environment Options form for the UltraSim circuit simulator.

➤ Select one of the following *Netlist Format* radio buttons:

- *spectre* specifies a Spectre format netlist
- *hspice* specifies an HSpice format netlist

Run Mode

The *Run Mode* radio buttons appear on the Environment Options form for UltraSim and UltraSimVerilog circuit simulators.

➤ Select one of the following *Run Mode* radio buttons:

- *Batch*
- *Interactive*

  If you select *Interactive*, you must type a valid name in the *Interactive Control File* field.

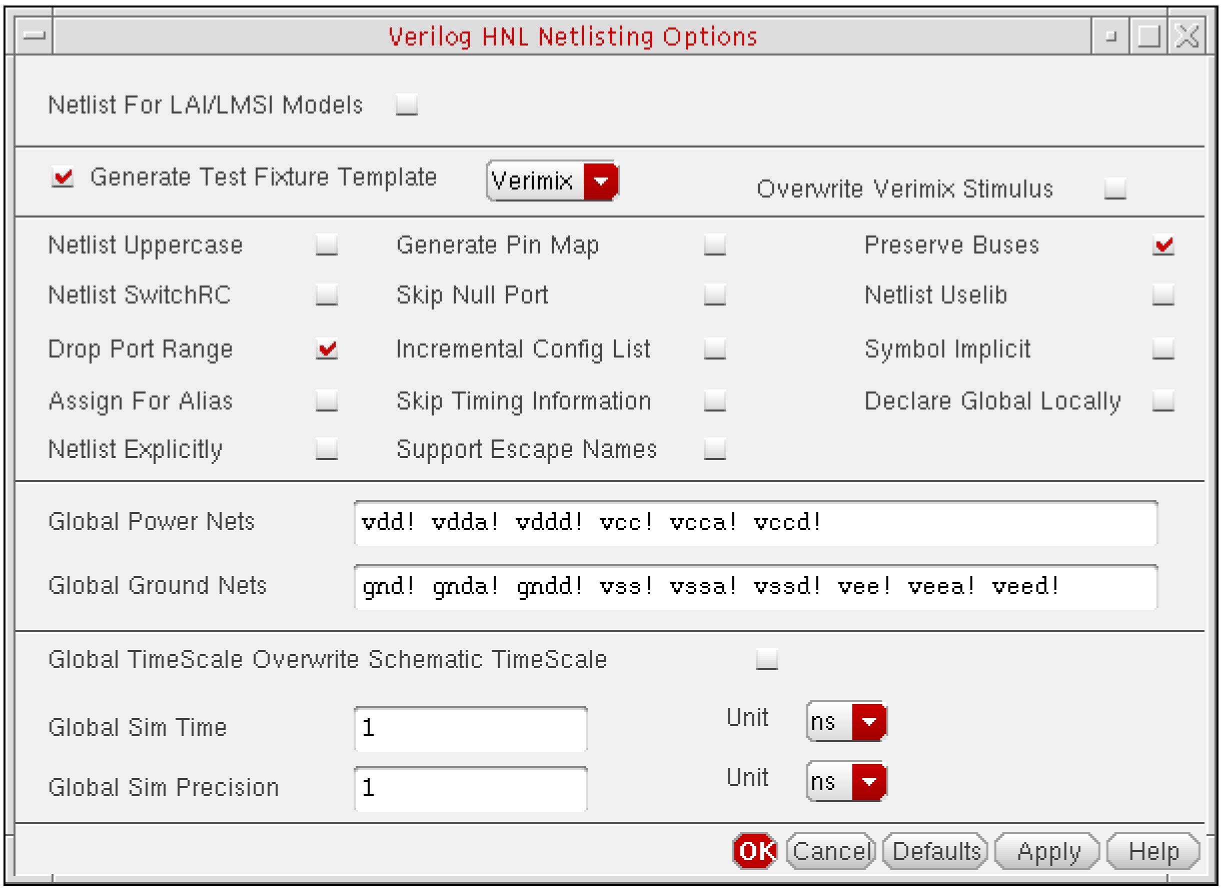
Verilog Netlist Option

The *Verilog Netlist Option* button appears on the Environment Options form for SpectreVerilog and UltraSimVerilog circuit simulators.

To specify Verilog netlisting options, do the following:

➤ Click *Verilog Netlist Option*.

The Verilog HNL Netlisting Options form appears.



For more information about this form, see the following:

- ❑ "Netlisting Options" in the "UltraSimVerilog" chapter of the *Virtuoso Analog Design Environment L User Guide*. (6.1.6 ONLY)
- ❑ "Verilog Netlisting Options" in the "Netlisting Options" chapter of the *Virtuoso Mixed-Signal Circuit Design Environment User Guide*.
- ❑ "Verilog Netlist Setup Form" in the *Virtuoso NC Verilog User Guide*

Specifying Simulation Options

The options you can specify depend on your target simulator.

- Specifying Simulator Options for Spectre, SpectreVerilog, UltraSim, and UltraSimVerilog Circuit Simulation on page 119
- Specifying Options for AMS Circuit Simulation on page 120
- Specifying Options for hspiceD Circuit Simulation on page 121

Specifying Simulator Options for Spectre, SpectreVerilog, UltraSim, and UltraSimVerilog Circuit Simulation

Important

The SpectreVerilog and UltrasimVerilog circuit simulators are available only in IC6.1.6 release.

To specify simulator options for Spectre, APS, SpectreVerilog, UltraSim, or UltraSimVerilog circuit simulation, do the following:

1. On the Data View assistant pane, right-click the test and choose one of the following:

Menu Choice	Form
Options – Analog	The Simulator Options form appears
Options – Digital	The Verilog-XL Simulation Options form appears. **Note:** This menu is available for SpectreVerilog and UltraSimVerilog circuit simulators only.
Options – Mixed Signal	The Mixed Signal Options form appears. **Note:** This menu is available for SpectreVerilog and UltraSimVerilog circuit simulators only.

   The fields that appear on the form depend on your target simulator. Refer to your simulator's user guide for information about the various options you can select.

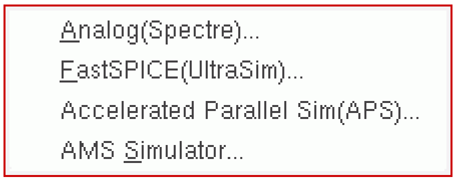
2. When you are done specifying options, click *OK*.

   The simulator options you specified are applied to future simulations.

Specifying Options for AMS Circuit Simulation

To specify options for AMS circuit simulation, do the following:

1. On the Data View assistant pane, right-click the test, choose *Options*, then choose one of the following:



   - *Options – Analog (Spectre)*
   - *Options – AMS Simulator*

   The appropriate form appears.

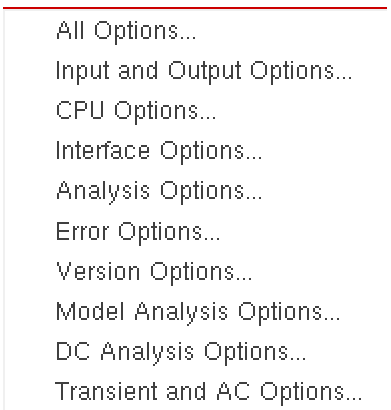
2. When you are done specifying options, click *OK*.

   The simulator options you specified are applied to future simulations.

Specifying Options for hspiceD Circuit Simulation

To specify options for hspiceD circuit simulation, do the following:

1. On the Data View assistant pane, right-click the test, choose *Options*, then choose one of the following:



   - *Analog Options – All Options*
   - *Analog Options – Input and Output Options*
   - *Analog Options – CPU Options*
   - *Analog Options – Interface Options*
   - *Analog Options – Analysis Options*
   - *Analog Options – Error Options*
   - *Analog Options – Version Options*
   - *Analog Options – Model Analysis Options*
   - *Analog Options – DC Analysis Options*
   - *Analog Options – Transient and AC Options*

   The appropriate form appears.

2. When you are done specifying options, click *OK*.

   The simulator options you specified are applied to future simulations.

Creating and Viewing the Netlist

To create a netlist, do the following:

1. On the Data View assistant pane, right-click the test and choose *Netlist – Create*.

   The netlist appears in a text browser window.

2. When you are done viewing the netlist, choose *File – Close Window*.

To recreate the netlist, do the following:

1. On the Data View assistant pane, right-click the test and choose *Netlist – Recreate*.

   A message appears in your Command Interpreter Window indicating success or failure.

To view the netlist, do the following:

1. On the Data View assistant pane, right-click the test and choose *Netlist – Display*.

   The netlist appears in a text browser window.

2. When you are done viewing the netlist, choose *File – Close Window*.

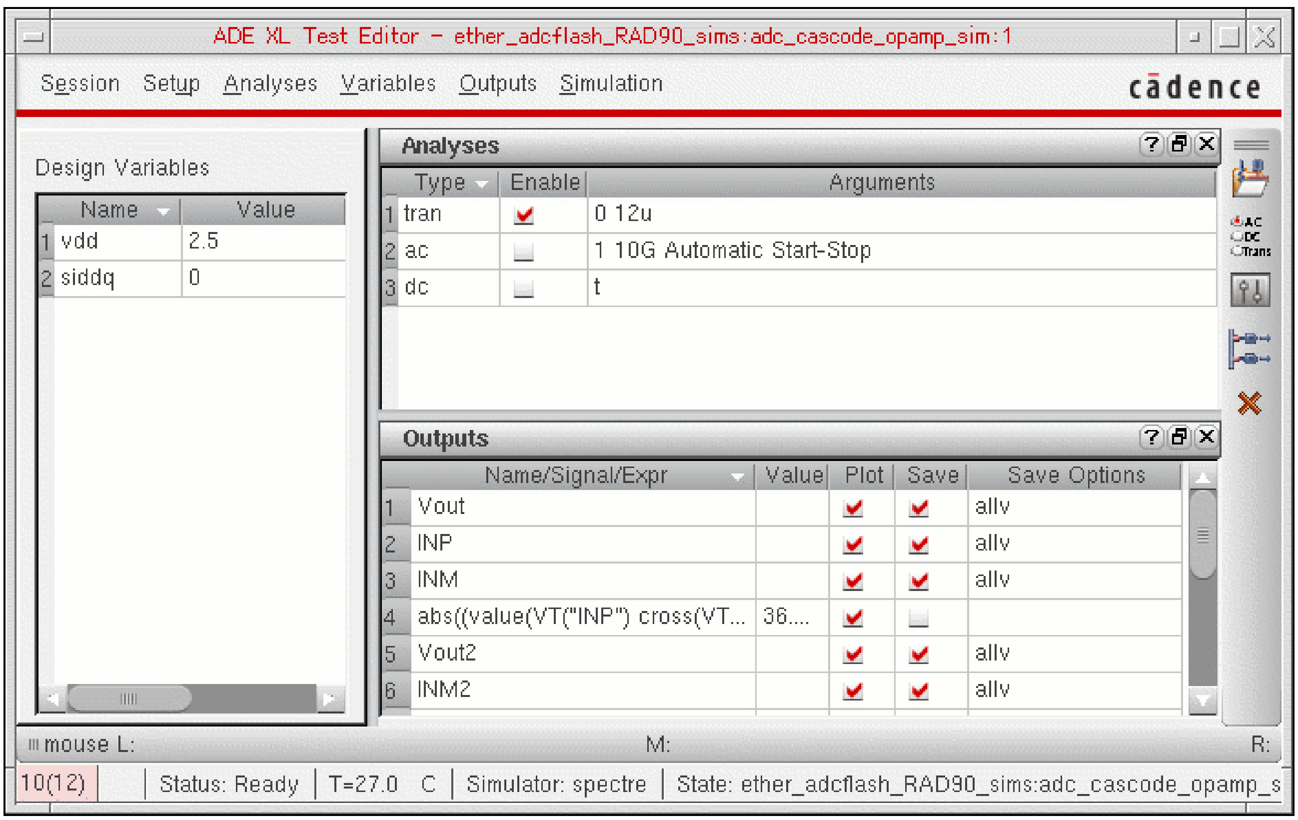
Modifying a Test in the Test Editor Window

As an alternative to using the ADE XL environment, you can use the ADE XL Test Editor to edit an ADE XL test. In this editor, you can choose a design, update design variables, set up analyses, and specify output variables.

To open the ADE XL Test Editor window for a test, do the following:

➤ On the *Tests* tree in the Data tab of Data View assistant pane, double-click a test name.

The ADE XL Test Editor window appears.



Alternatively, you can do the following:

➤ Right-click the test name and choose *Open Test Editor*.

The ADE XL Test Editor window appears.

This window displays the test details in the same way as displayed in the ADE L environment. Here, you can create and modify the setup. If required, you can also run simulations and view output results and plots.

Similar to the ADE L environment, the Test Editor contains the *Save By Subckt Instances* pane. Settings done in this pane are used by the ADE XL simulations, but currently these settings cannot be viewed in the ADE XL environment. For details on the *Save By Subckt Instances* pane, refer to the *Specifying Hierarchy Levels to Save Options* section in the Analog Design Environment L User Guide.

Important

By default, you cannot run simulations in the ADE XL Test Editor. However, if required, you can set the showAllMenus environment variable to `t`. When this variable is set, all menus that are used to run simulations appear in the Test Editor window.

When you run simulations, same simulation settings that were set in the ADE XL environment are used. However, an important difference is that simulations that run from the test editor do not use ICRP. They are same as ADE L simulations. That is why, for a distributed simulation, only relevant job settings are copied from the ADE XL environment.

Note: When you run simulations from the test editor, by default, the results are saved in the same directory as that of the previous history run. This overwrites the simulation data saved for the previous history run. If you want to save data for the history, set the debugDataDir environment variable to specify a different location where you want to save results for the simulations run from the ADE XL Test Editor.

When you close the Test Editor window, any changes done here are copied back to the ADE XL setup.

Important

If *Reliability Analysis* was not enabled in the adexl view and you enable it for a test in the test editor, after you close the Test Editor, a default reliability setup named `Default_Relx` is created in the Data View pane in the ADE XL window.

For information about using the Test Editor window, see “Environment Setup” in the *Virtuoso Analog Design Environment L User Guide*.

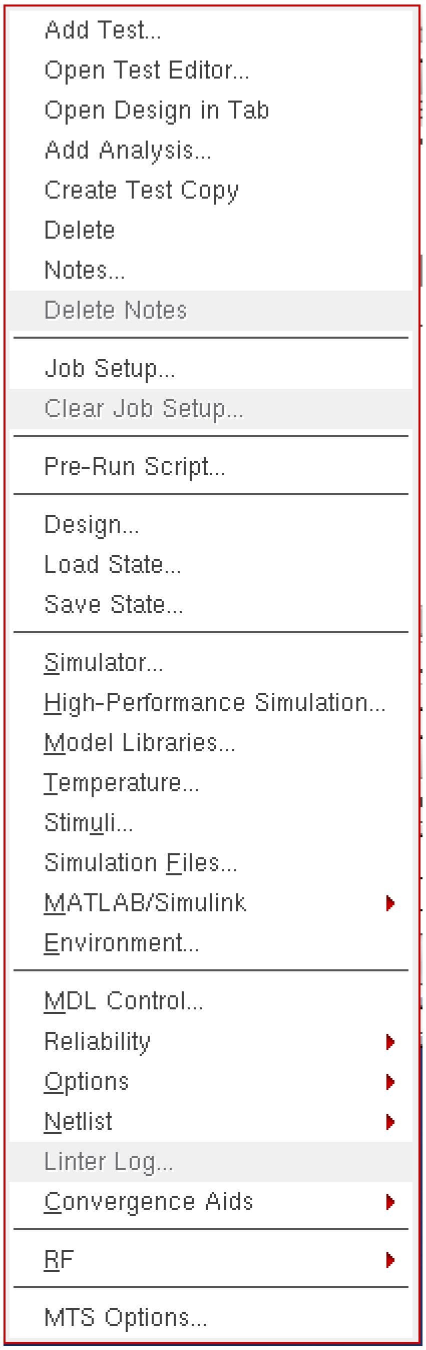
Working with OCEAN-Based Tests

In ADE XL, you can save the simulation setup in OCEAN scripts and run these scripts either from UNIX shell or from Virtuoso GUI.

For more details on creating and running OCEAN scripts, refer to Creating and Running an OCEAN Script.

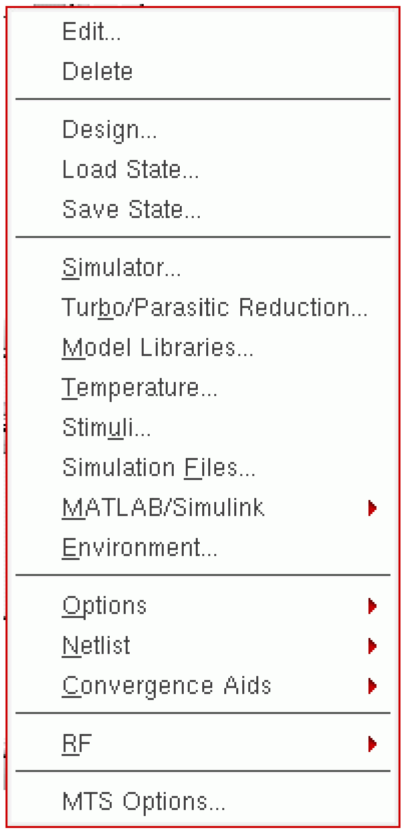
Data View Assistant Right-Click Menus

When you right-click a test name on the Data View assistant pane, the following pop-up menu appears:



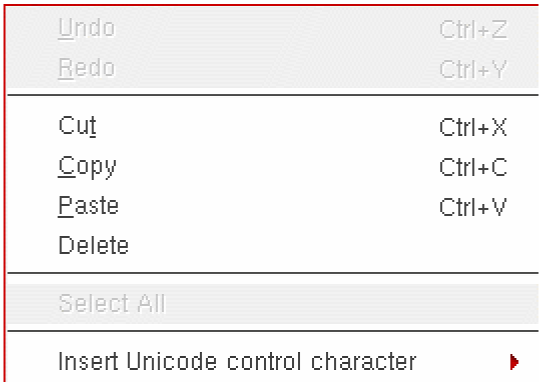
Note: The *MTS Options* item appears only after you have launched Analog Design Environment GXL.

When you right-click an analysis on the Data View assistant pane, the following pop-up menu appears:

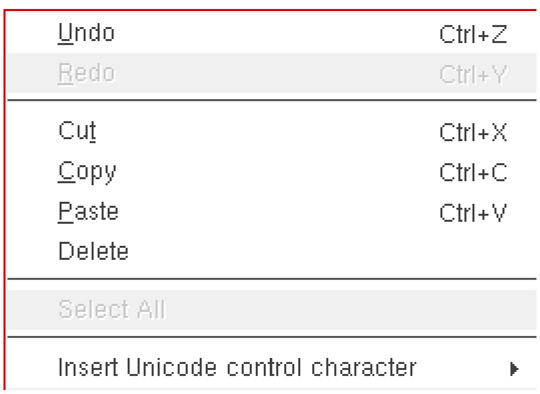


Note: The *MTS Options* item appears only if you have launched Analog Design Environment GXL.

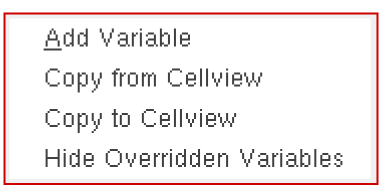
When you right-click in edit mode for a test name on the Data View assistant pane, the following pop-up menu appears:



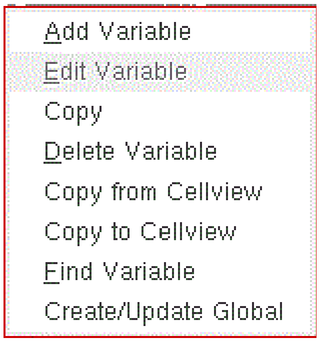
The action you take determines what menu items are available.



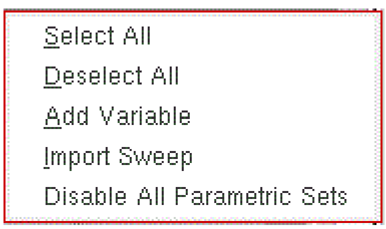
When you right-click the *Design Variables* tree on the Data View assistant pane, the following pop-up menu appears:



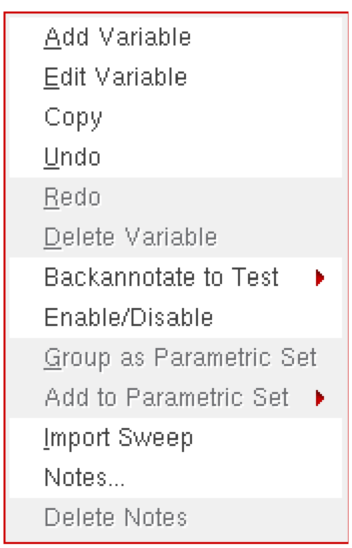
When you right-click a design variable on the Data View assistant pane, the following pop-up menu appears:



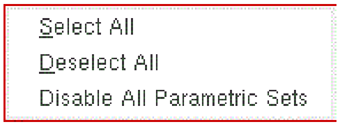
When you right-click the *Global Variables* tree on the Data View assistant pane, the following pop-up menu appears:



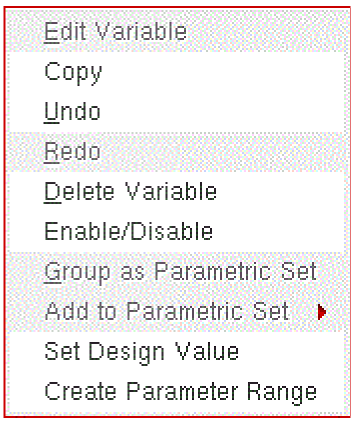
When you right-click a global variable on the Data View assistant pane, the following pop-up menu appears:



When you right-click the *Parameters* tree on the Data View assistant pane, the following pop-up menu appears:



When you right-click a parameter on the Data View assistant pane, the following pop-up menu appears:



3

Working with Design Variables and Instance Parameters

You can use design variables and Component Description Format (CDF) instance parameters to set component values for simulation. Design variable values apply to specific tests. The scope of a CDF parameter value depends on which Analog Expression Language (AEL) functions you use to refer to the parameter.

See the following topics for more information:

Working with Design Variables

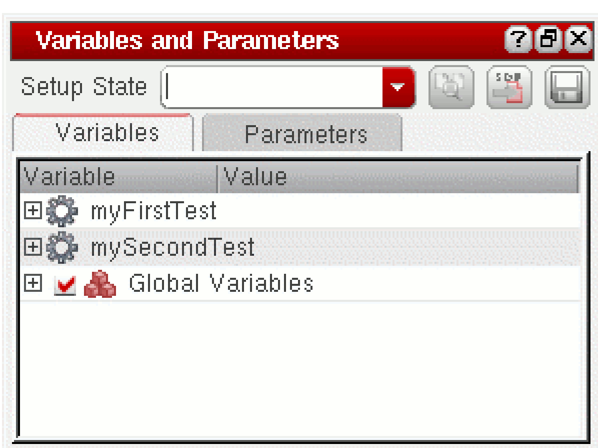
You can add, change, and delete design variables on the Data View pane or the Variables tab of the Variables and Parameters pane in your ADE XL environment.

Adding a Design Variable

To add a design variable to a test, do the following:

1. Choose *Window – Assistant – Variables and Parameters*.

   The Variables and Parameters pane appears.



2. On the Variables tab of the Variables and Parameters pane, right-click the test and choose *Add Variable*.

Tip

Alternatively, you can do one of the following:

- ❑ Click where it says *Click to add variable* in an expanded test tree on the Data View pane or on the Variables tab of the Variables and Parameters pane.

Figure 3-1 Design Variables in the Data View Pane

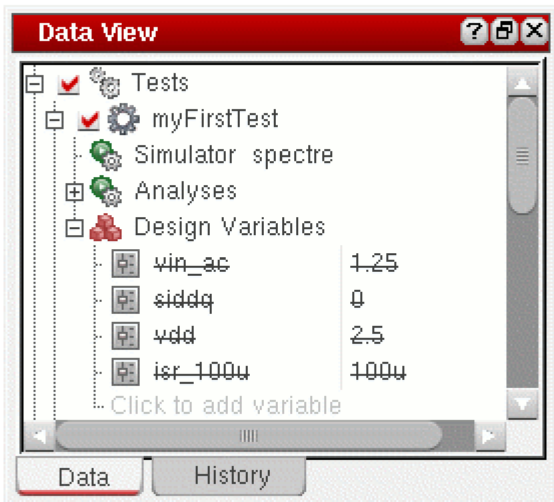
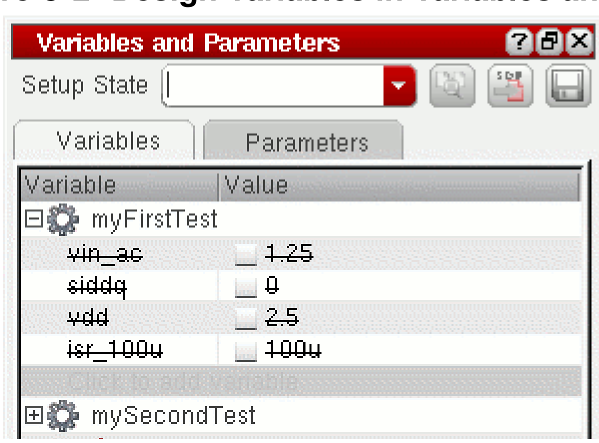
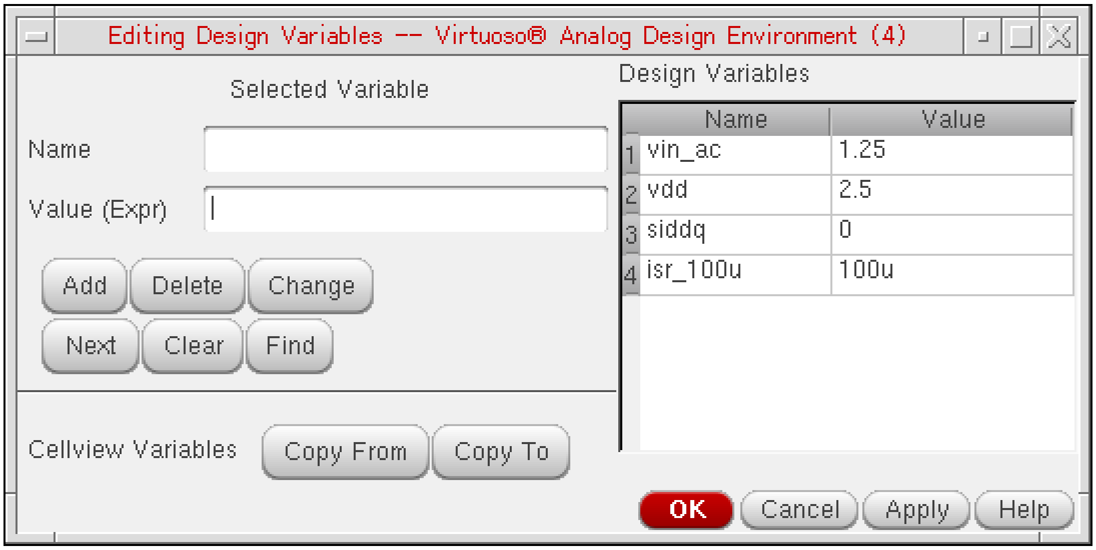


Figure 3-2 Design Variables in Variables and Parameters Assistant



- ❑ In an expanded test tree on the Data View pane, right-click a design variable and choose *Add Variable*.

The Editing Design Variables form appears.



3. In the *Name* field, type a name for your design variable.

   The name must begin with a letter or the underscore character (_) and can contain only letters, numbers, or underscore characters.

4. In the *Value (Expr)* field, type a value or expression for your design variable.

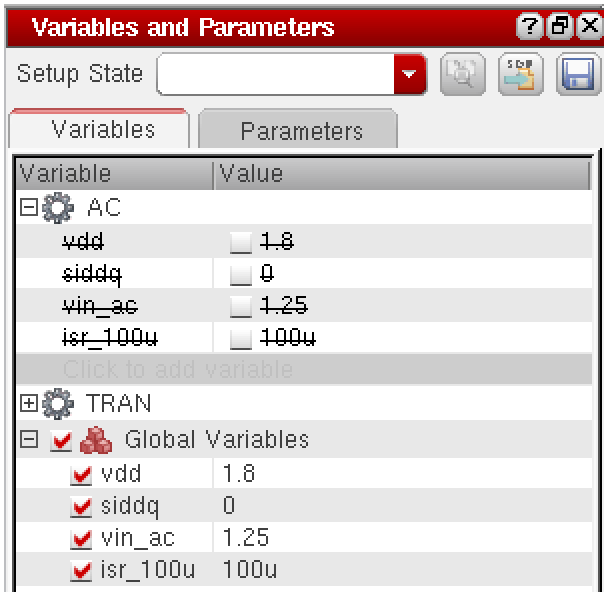
   The expression can be an equation, a function, or another variable. The simulator evaluates the expression which must follow Analog Expression Language (AEL) syntax.

5. Click *Add*.

   Your design variable and its value appear in the *Design Variables* table on the Editing Design Variables form and also in the expanded test tree on the Data View pane and on the Variables tab of the Variables and Parameters pane.

   Note: You can also define variables in a definitions file.

   By default, the design variables defined for your tests are automatically promoted as global variables in the *Global Variables* tree on the Data View assistant pane and the Variables tab of the Variables and Parameters assistant pane. The design variables are displayed with a ~~strikethrough~~ because the settings for a global variable override the settings of design variables that have the same name as the global variable. For example, in the following figure, the design variables specified for the test named `myFirstTest` are automatically added as global variables. The design variables are displayed with a ~~strikethrough~~ because the global variables override the design variables.



Note: You can disable the default automatic copy of design variables to global variables by setting the autoPromoteVarsToGlobal environment variable to `nil`.

Changing the Value of a Design Variable

You can change the values of design variables that are not overridden by global variables.

If you want to change the value of a design variable that is overridden by a global variable, you must do one of the following:

- Disable the global variable for test in which you want to change the value of the design variable.

  For more information, see Disabling Global Variables for Specific Tests on page 170.

- Disable the global variable for all tests.

  For more information, see Enabling and Disabling Global Variables for All Tests on page 168.

To change a design variable, do the following:

1. In an expanded test tree on the Variables tab of the Variables and Parameters pane, double-click in the *Value* cell of the design variable you want to change.

Tip

Alternatively, in an expanded test tree on the Data View pane, double-click on the value of the design variable you want to change.

2. Type a new value or expression for the design variable.
3. Press *Return* or click anywhere outside that table cell.

   The new value appears in the *Value* cell for that design variable.

Alternatively, you can change the value of a design variable from the Editing Design Variables form as follows:

1. In an expanded test tree on the Data View pane or on the Variables tab of the Variables and Parameters pane, right-click the design variable and choose *Edit Variable*.

   The Editing Design Variables form appears.

2. In the *Design Variables* table, select the design variable you want to change.

   The row appears highlighted and the *Selected Variable* fields, *Name* and *Value (Expr)*, contain the name and value of the design variable you selected.

3. Click in the *Value (Expr)* field.

   The editing cursor appears where you clicked.

4. Type a new value or expression.
5. Click *Change*.

   The new value appears in the *Design Variables* table.

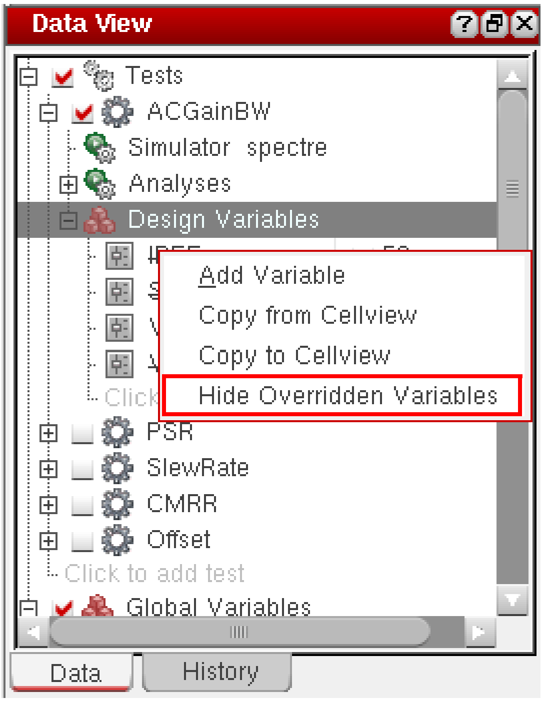
6. Click *OK*.

Hiding Overridden Design Variables for a Test

By default, a design variable for a test is automatically promoted as global variables and the variable appears with a strikethrough in the list of variables for that test.

You can choose to hide all the overridden variables for a test and to display only those variables for which the value specified for a test is used. For this, do any one of the following:

➡ In the Data View pane, expand the design tree, right-click on *Design Variables* and choose *Hide Overridden Variables*.



➡ On the Variables tab of the Variables and Parameters assistant, right-click on a test name and choose *Hide Overridden Variables*.

In both the Data View pane and the Variables and Parameters assistant, the overridden variables are hidden for the test.

Deleting a Design Variable

To delete a design variable, do the following:

➤ In an expanded test tree on the Data View pane or on the Variables tab of the Variables and Parameters pane, right-click the design variable you want to delete and choose *Delete Variable*.

Note: You can also delete design variables from the Editing Design Variables form.

Saving Design Variables

Your design variables are saved with your ADE XL setup. To perform an explicit save, do the following:

➤ Choose *File – Save*.

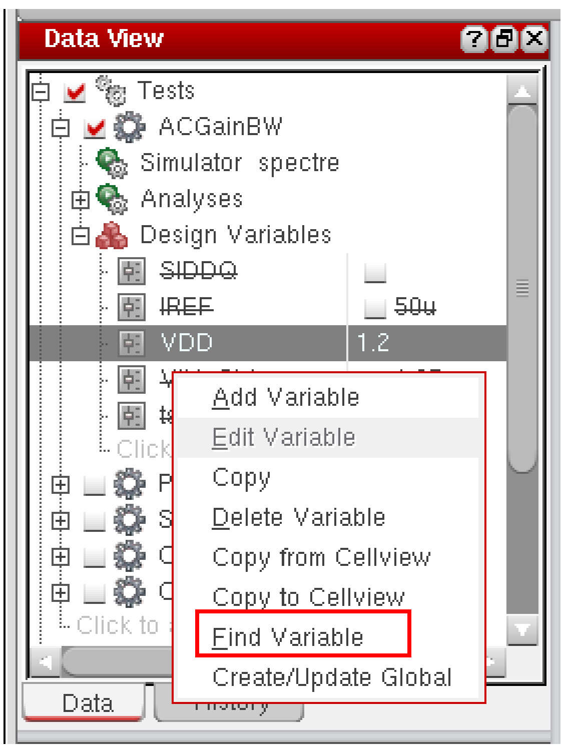
The program saves your setup information, including design variables. The program reloads these values the next time you open this design cellview.

Finding Design Variables in Schematic

You can search for a design variable in the schematic to identify its location. To find a variable, do the following:

1. In the Data View pane, expand the test tree.
2. Expand the Design Variables tree.

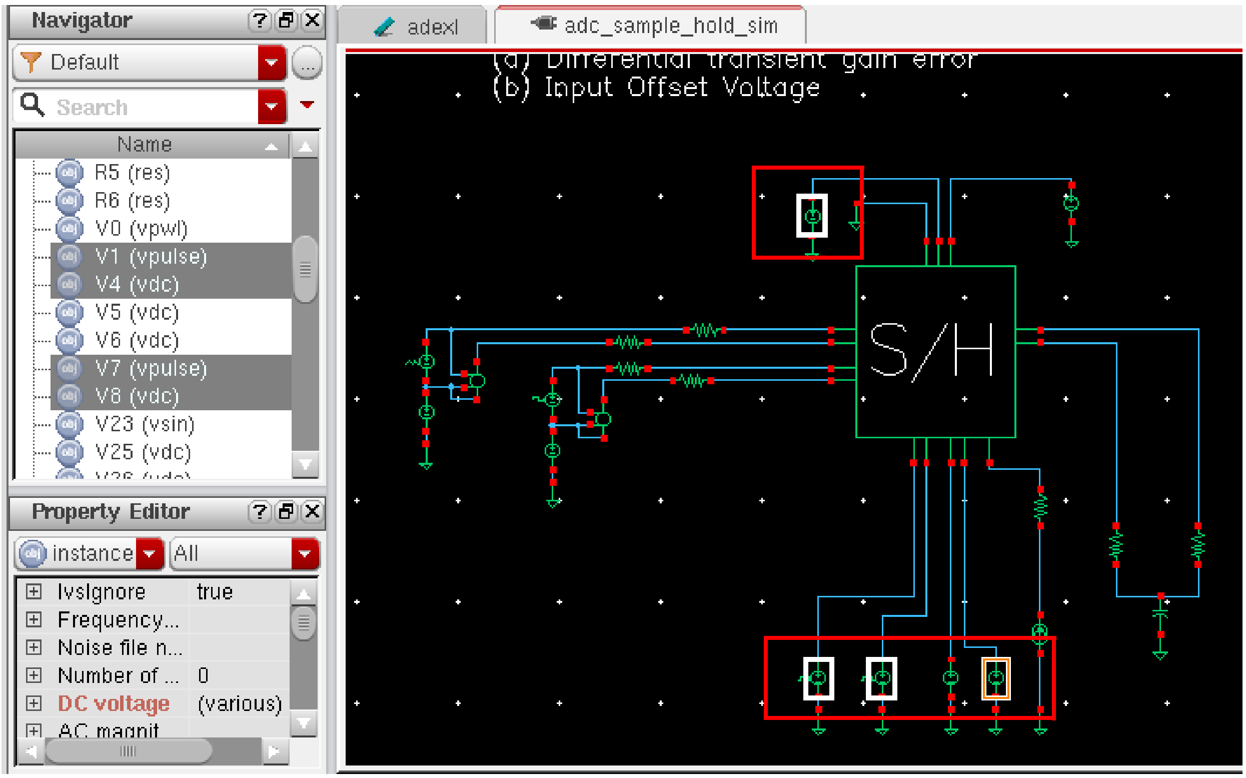
Right-click a design variable and choose *Find Variable*.



Alternatively, do the following on the Variables tab of the Variables and Parameters assistant:

1. Toggle the view to show the variables names. Expand a variable tree to show the test names.
2. Right-click the test name and choose *Find Variable*.

If the variable is found in the schematic, the design is opened in the design tab and all the instances are highlighted, as shown below.



In case of a hierarchical design, if the variable is not found in the top level of design, the tool traverses down the hierarchy and highlights the variable in the lower level, where found.

If the variable is not found in the schematic, a warning message, as shown below, is displayed in the CIW.

```
Variable '<var-name>' not found on the schematic.
```

Copying Design Variable Values to the Schematic

If you change design variables in your simulation setup and want to copy the values back to the cellview before you save the schematic, do the following:

➤ On the Variables tab of the Variables and Parameters pane, right-click the test and choose *Copy to Cellview*.

Tip

Alternatively, in an expanded test tree on the Data View pane, right-click a design variable and choose *Copy to Cellview*.

The program copies design variable values from the Variables tab of the Variables and Parameters pane to the cellview.

Alternatively, you can do the following:

1. On the Variables tab of the Variables and Parameters pane, right-click the test and choose *Edit Variable*.

Tip

Alternatively, in an expanded test tree on the Data View pane, right-click a design variable and choose *Edit Variable*.

The Editing Design Variables form appears.

2. Select one or more variables you want to copy to the current cellview.

3. In the *Cellview variables* section, click *Copy To*.

   The program copies the selected design variable values to the current cellview.

4. When you are done with the form, click *OK*.

Copying Design Variable Values from the Schematic

If you change design variables on your schematic and want to use these values in your next simulation, do the following:

➤ On the Variables tab of the Variables and Parameters pane, right-click the test and choose *Copy from Cellview*.

Tip

Alternatively, in an expanded test tree on the Data View pane, right-click a design variable and choose *Copy from Cellview*.

The program copies design variable values from the schematic to the Variables tab of the Variables and Parameters pane.

Alternatively, you can do the following:

1. On the Variables tab of the Variables and Parameters pane, right-click the test and choose *Edit Variable*.

Tip

Alternatively, in an expanded test tree on the Data View pane, right-click a design variable and choose *Edit Variable*.

The Editing Design Variables form appears.

2. Click *Copy From*.

   The program copies design variable values from the schematic to the Variables tab of the Variables and Parameters pane.

3. When you are done with the form, click *OK* or *Cancel*.

Note: You can disable copying design variables to global variables by setting the `copyDesignVarsFromCellview` environment variable to `nil`.

Importing Design Variables from an ADE State

For details on importing the design variables and parameters by using the **Loading State** form, see "Loading State Information" on page 92.

Displaying Design Variables on the Schematic

To display the values of instance parameters that are design variables on the schematic, do the following:

1. In the CIW, choose *Tools – CDF – Edit*.

   The Edit CDF form appears.

2. In the *Scope* group box, select the *Library* radio button.

3. In the *CDF Layer* group box, select one of the following CDF types (see *CDF Type* in the CDF Commands chapter in the *Component Description Format User Guide* for information about these choices):

   - *Effective*
   - *User*

- ❑ *Base*

4. In the *Library Name* field, select the library whose CDF you want to edit.
5. On the Edit CDF form, click the *Interpreted Labels* tab.

   You can use interpreted labels to display parameter values, evaluated parameter values, net connectivity information, backannotated simulation information, and more. For more information, see “Specifying Label Information” in the *Component Description Format User Guide*.
6. On the Interpreted Labels tab, click the *Parameters(cdsParam)* tab.
7. For *paramEvaluate*, select the *full* check box.

   Design variables appear on your schematic.

Defining Variables in a File

You can define functions and global variables that are not design variables (such as model parameters and simulator parameters) in a definitions file. Cadence provides sample definition files for the Spectre circuit simulator in `your_install_dir/tools/dfII/samples/artist/models/spectre` (see `defaults.scs` and `definitions.scs`).

To set up a definitions file, do the following:

1. Create the file in the directory you specify in the *Include Paths* field on the Simulation Files Setup form.
2. In the *Definition Files* field on the Simulation Files Setup form, type the full UNIX path and name of the definitions file.

The simulator reads the definitions file when it starts.

Your definitions file can contain any of the following:

Simple passing parameters	`real R(real l, real w) {`
	`return (500*l/w);`
	`}`

Functions returning constant values	`real PiRho() {`
	`return 2500;`
	`}`
	`real Rpi(real l, real w) {`
	`return PiRho()*l/w;`
	`}`

For more information about defining functions in a definitions file, see “User Defined Functions (functions)” in the *Virtuoso Spectre Circuit Simulator Reference*.

See also

- Example of a Polynomial Resistor as a Function of Temperature on page 143
- About Inherited Parameter Value Functions (iPar, pPar) on page 143

Example of a Polynomial Resistor as a Function of Temperature

You can define a polynomial resistor as a function of temperature as follows:

```
real rpoly(real value, real tdc) {
value*(1+.01*(tdc-25)+.002*(tdc-25)**2);
}
```

You can use this function to specify the value of a resistor as follows:

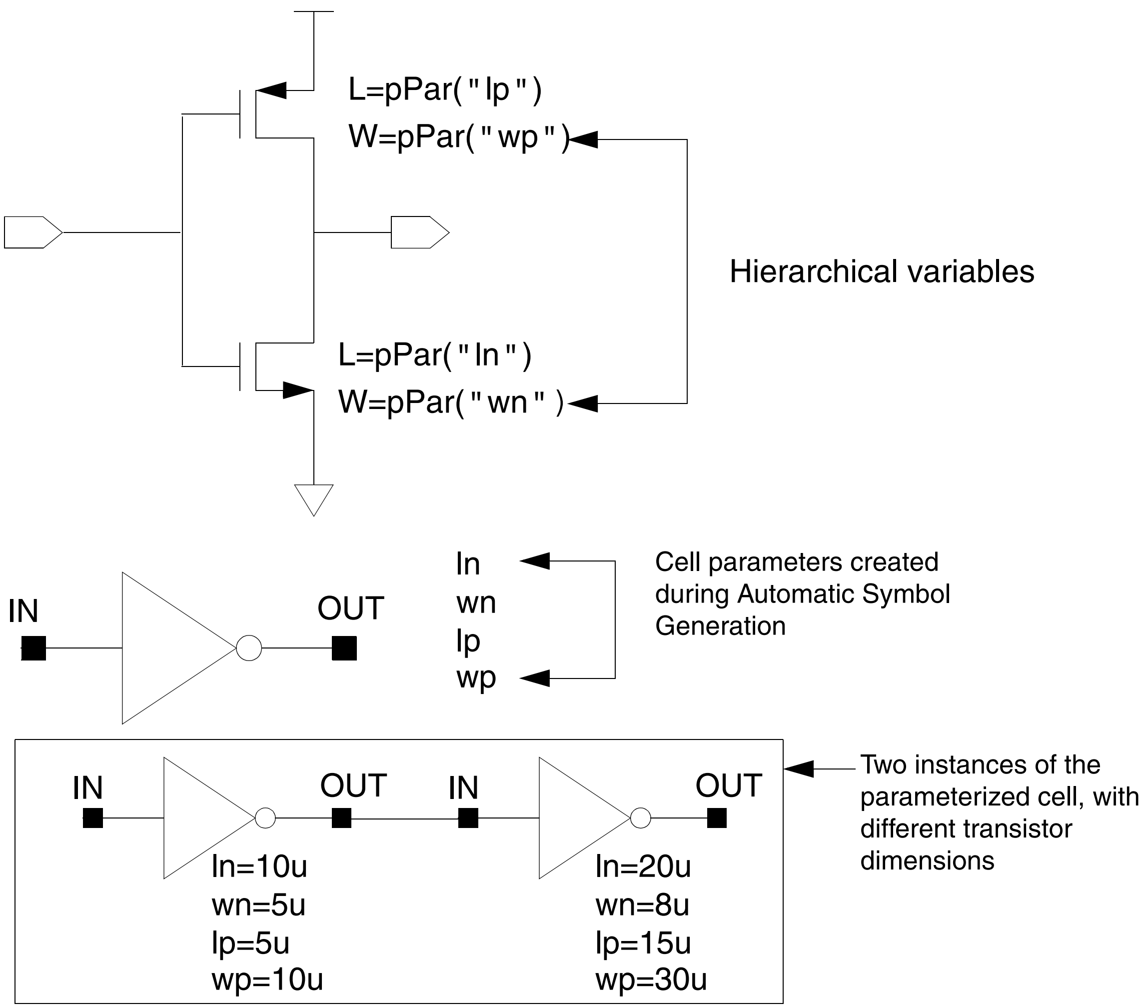
```
rpoly(1k,tempdc)
```

You can set resistor properties `tc1` and `tc2` so that the system automatically models resistor temperature effects, rather than defining your own functions.

About Inherited Parameter Value Functions (iPar, pPar)

You can use inherited parameter value functions iPar and pPar in conjunction with built-in functions or user-defined functions.

The following diagram shows a schematic example using pPar:



For more information about iPar and pPar parameters, see “Passing Parameters in a Design” in the <u>Defining Parameters</u> chapter of the <u>*Component Description Format User Guide*</u>.

Working with Device Instance Parameters

In ADE XL, you can

- Change the value of a device instance parameter for a simulation without affecting the value on the schematic
- Disable the changed value of a device instance parameter for a simulation
- Sweep a device instance parameter
- Disable callbacks on swept device parameters
- Create matched device parameters
- Create ratio-matched device parameters
- Backannotate changed device instance parameter values back to the schematic

See the following topics for more information:

- Viewing Instance Parameters and Their Values on page 146
- Changing the Value of a Device Instance Parameter for Simulation on page 149
- Disabling a Changed Device Instance Parameter Value for a Simulation on page 152
- Filtering Device Instance Parameters on page 152
- Creating Custom Device Filters on page 153
- Deleting a Parameter on page 155
- Specifying an Instance Parameter as a Sweep Parameter on page 178
- Creating Parameter Ranges on page 156
- Creating Matched Device Parameters on page 201
- Creating Ratio-Matched Device Parameters on page 204
- Sorting Parameters by Properties and Objects on page 210
- Disabling Callbacks on Swept Device Parameters on page 212
- Annotating Simulation Results on page 693

Viewing Instance Parameters and Their Values

To view instance parameters and their values for the design associated with your test, do the following:

1. Make sure you are viewing the design schematic associated with your test.

   For more information, see Opening a Design Schematic on page 90.

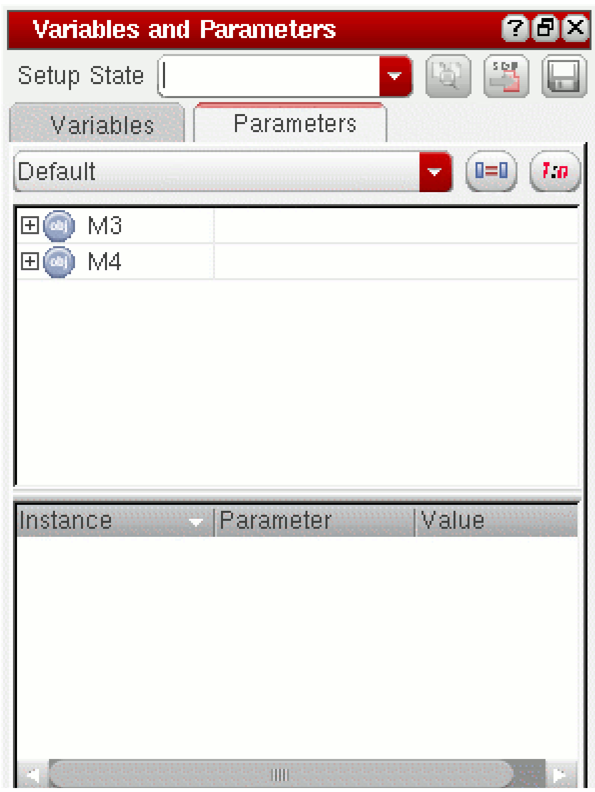
2. Choose *Window – Assistant – Variables and Parameters* to display the Variables and Parameters assistant pane.
3. Click the Parameters tab on the Variables and Parameters assistant pane.
4. On the schematic, select one or more instances.

   You can select more than one instance by shift-clicking each instance one by one or by dragging a selection box around a set of instances.

Tip

Use the Navigator assistant pane to quickly select the instances.

The selected instances are displayed in the upper half of the Parameters tab of the Variables and Parameters pane.

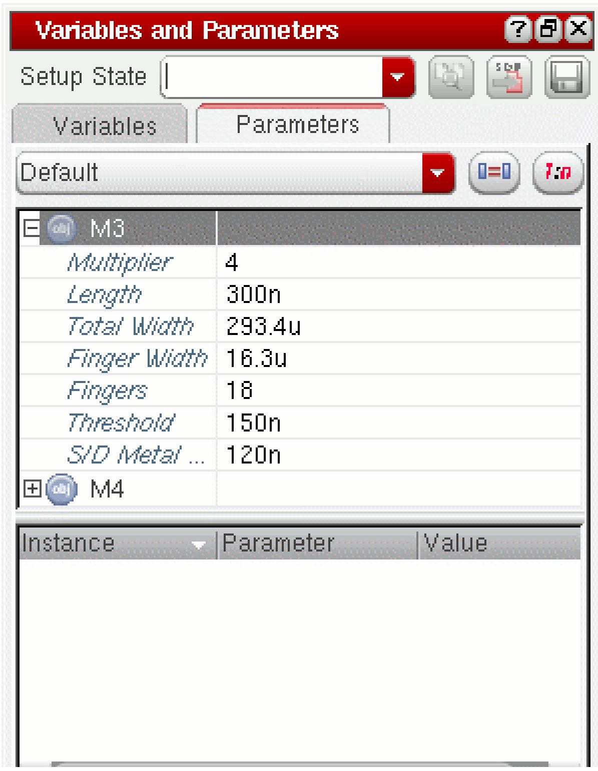


Tip

Place the mouse pointer on the instance name to view the name of the library, cell, and view containing the instance in a pop-up.

5. Click the plus sign to the left of the instance name to view the parameters for the instance.

The parameter names appear in the expanded tree beneath the instance name. The value of each parameter appears in the second column.


Variables and Parameters
Setup State
Variables
Parameters
Default
M3
Multiplier
4
Length
300n
Total Width
293.4u
Finger Width
16.3u
Fingers
18
Threshold
150n
S/D Metal ...
120n
M4
Instance
Parameter
Value

Changing the Value of a Device Instance Parameter for Simulation

To change the value of a device instance parameter for simulation, do the following:

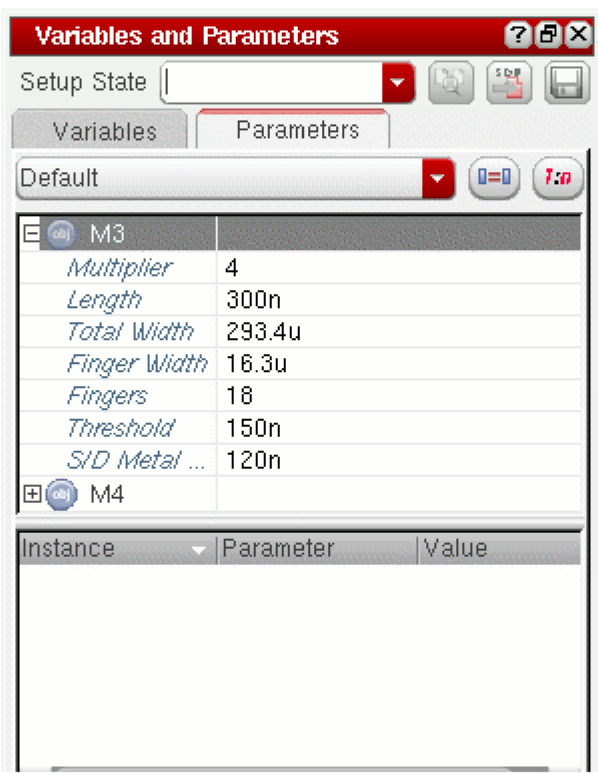
1. On the upper half of the Parameters tab on the Variables and Parameters pane, click the value of the device instance parameter you want to change and type a new value.

   Note: You might have to expand the tree by clicking the plus sign to the left of the instance name at the root of the tree.

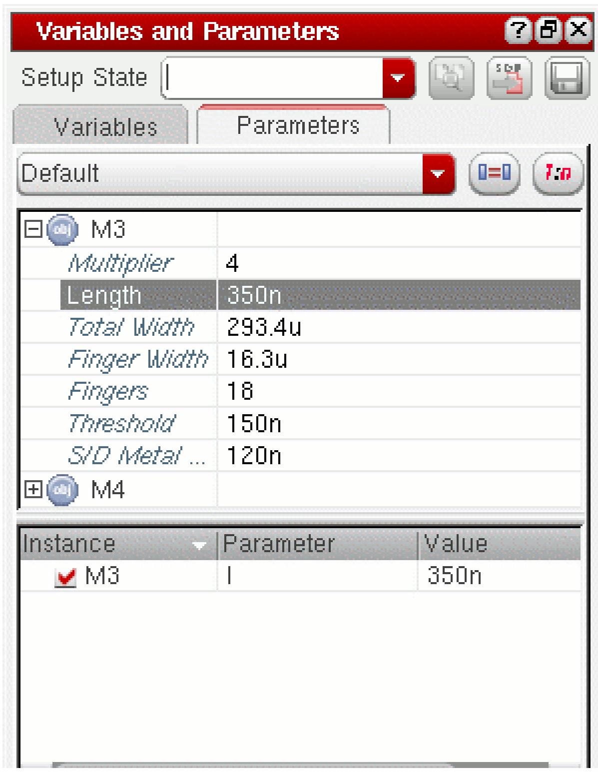
Tip

To view only the parameters in the Parameters tab, see Sorting Parameters by Properties and Objects on page 210.

For example, change the value of the *Length* parameter from `300n` to `350n`.



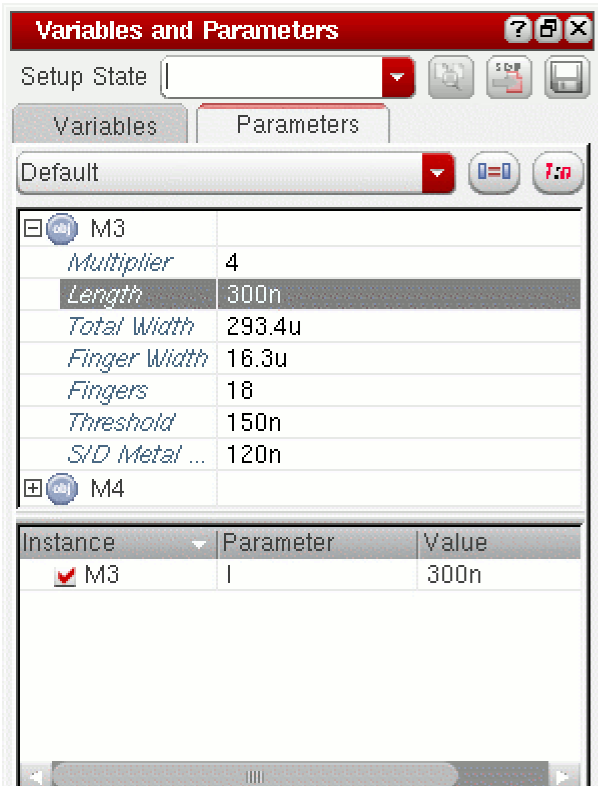
The instance parameter and its new value (`350n`) appear on the lower half of the Parameters tab on the Variables and Parameters assistant pane. The simulator uses this value when you click the Run Simulation button.



Alternatively, you can do the following:

1. Right-click the instance parameter whose value you want to change and choose *Create Parameter*.

   The instance parameter and its value appear on the lower half of the Parameters tab on the Variables and Parameters assistant pane.



2. Double-click the value on the lower half of the Parameters tab and type a new one.

   The simulator uses this value when you click the Run Simulation button.

The parameter is also displayed in the *Parameters* tree in the Data View pane.

Note: You can add parameters only using the Parameters tab of the Variables and Parameters assistant pane. Parameters cannot be directly added in the Data View pane.

Tip

At any time, you can reset the original value, which was set in the design schematic, for a parameter. For this, in the lower half of the Parameters tab, right-click on a parameter name and choose *Set to Design Value*.

Also see:

- Enabling and Disabling Parameters
- Specifying an Instance Parameter as a Sweep Parameter
- Creating Parameter Ranges

Disabling a Changed Device Instance Parameter Value for a Simulation

To disable a changed device instance parameter value for a simulation, do the following:

➤ On the lower half of the Parameters tab on the Variables and Parameters assistant pane, deselect the check box for that parameter.

Tip

Alternatively, in an expanded *Parameters* tree on the Data View pane, deselect the check box for that parameter.

The next time you click the Run Simulation button, the simulator uses the device instance parameter value from the schematic.

Filtering Device Instance Parameters

To filter the device instance parameters that appear on the upper half of the Parameters tab on the Variables and Parameters assistant pane, do the following:

➤ In the *Filter* drop-down list, select one of the following:

CDF	All CDF parameters
CDF Editable	All editable CDF parameters
Default	All editable CDF parameters with non-nil CDF values
`customFilter`	Custom device instance parameter filter See “Creating Custom Device Filters” on page 153 for more information.

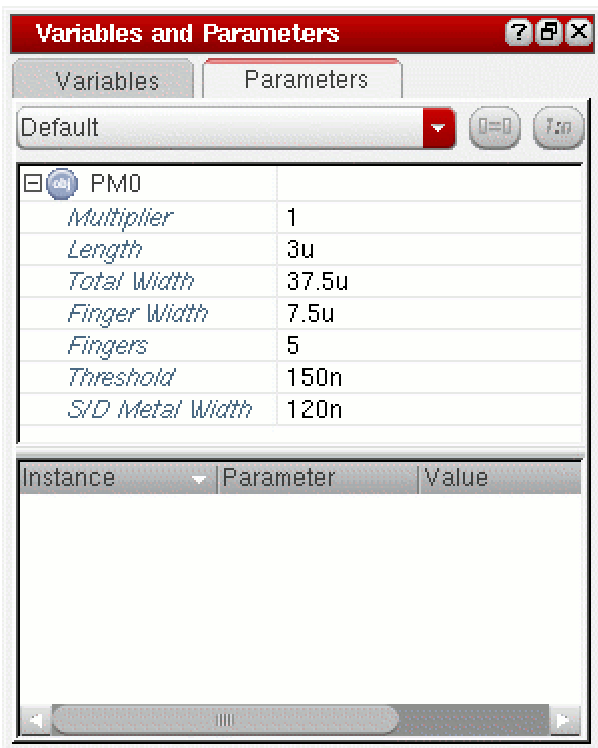
Only those device instance parameters that meet the filter specification appear on the upper half of the Parameters tab on the Variables and Parameters assistant pane.

Creating Custom Device Filters

To create a custom device instance parameter filter, do the following:

1. Write a procedure to define your custom filter.

   For example, if the *Default* filter displays the parameters for instance `PM0` of the `pmos2v` cell as shown below, and you want to filter out the `Total Width`, `Threshold` and `S/D Metal Width` parameters for all instances of the `pmos2v` cell, write the following procedure:




```
(procedure (myCustomFilter inst simulator)
    (let (paramList libName cellName instName)
    (setq instName inst->name)
    (setq cellName inst->cellName)
    (setq libName inst->libName)
    (setq paramList list())

    ;;Example filter for hiding Total Width, Threshold and S/D Metal Width
    ;;parameters for instances of cells whose names start with pmos2v
    (rexCompile "^pmos2v")
    (if (rexExecute cellName) then
         (setq paramList
```

```
                (list
                     (list "fw" (get inst "fw"))
                     (list "l" (get inst "l"))
                     (list "fingers" (get inst "fingers"))
                     (list "m" (get inst "m"))
                )))

    paramList
    )
)
```

Note: You must specify the CDF parameter names and not the CDF prompt names in the procedure. For example, you must specify the CDF parameter name `fw` and not the CDF prompt name `Finger Width` in the procedure.

Tip

You can also write procedures to filter parameters of instances starting with specific names (for example, all instances starting with the name `PM`), or to filter parameters of instances of cells in a library (for example, filter parameters of all instances of the `pmos2v` cell in the `gpdk090` library).

2. Call `axlRegisterCustomDeviceFilter` as follows:

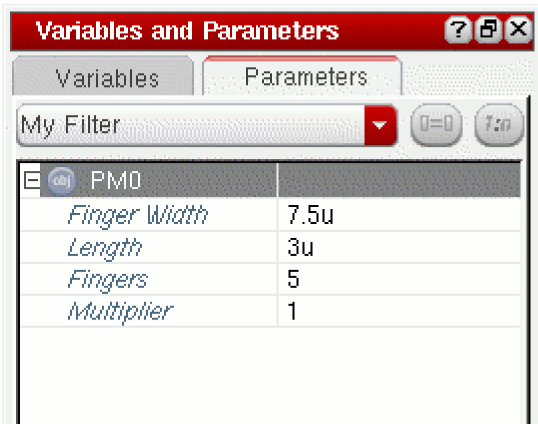
   ```
   axlRegisterCustomDeviceFilter "My Filter" 'myCustomFilter
   ```

   The function returns `t` if the registration is successful; otherwise, `nil`.

   My Filter appears in the *Filter* drop-down list.

To load your custom filters each time the program starts, you can declare and register them in your `.cdsinit` file.

When you select *My Filter* in the *Filter* drop-down list, the device parameters `Total Width`, `Threshold` and `S/D Metal Width` do not appear on the upper half of the Parameters tab on the Variables and Parameters assistant pane for the selected device instance(s).



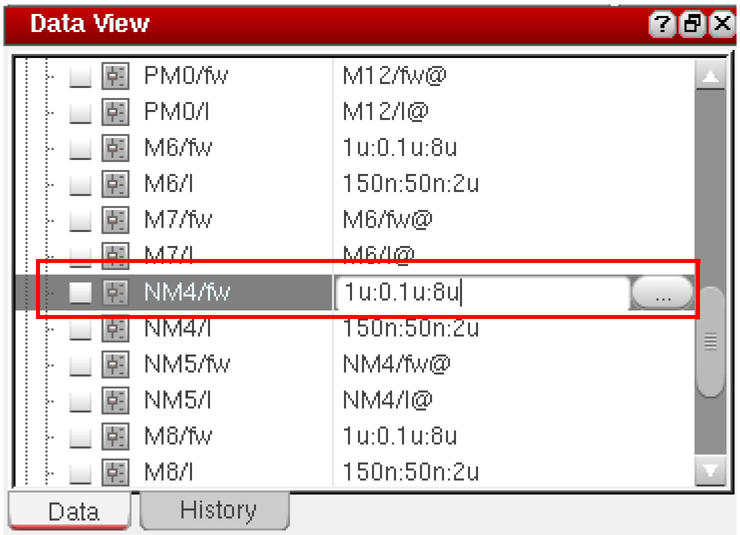
Deleting a Parameter

To delete a parameter, do the following:

➤ In an expanded *Parameters* tree on the Data View pane or on the lower part of the Parameters tab of the Variables and Parameters pane, right-click the parameter you want to delete and choose *Delete Parameter*.

Creating Parameter Ranges

You can specify a parameter range for an individual parameter by editing the value field in the Data View pane or the Variables and Parameters assistant, as shown below.



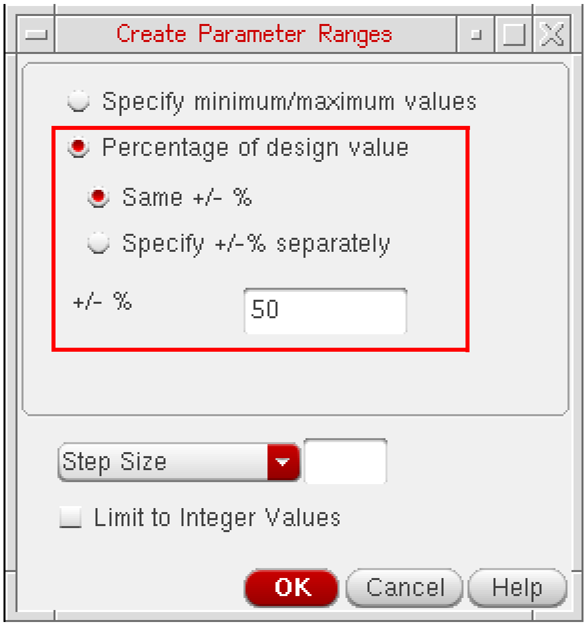
However, if the parameter list is long, applying a parameter range for each parameter individually can take a long time. Instead, you can use the *Create Parameter Range* command in the right-click menu of the Parameters list. This command applies a standard parameter range to a set of selected or all the parameters in the Parameters list.

To create a parameter range for more than one or all the parameters together, perform the following steps in Data View pane or the lower part of the Variables and Parameters assistant:

1. Select one or more variables in the Parameters list for which you want to create parameter range. Press the *Shift* or *Ctrl* key and select the parameter names.

2. Right-click on a parameter in the Parameters list and choose *Create Parameter Range*.

The Create Parameter Ranges form appears, as shown below.



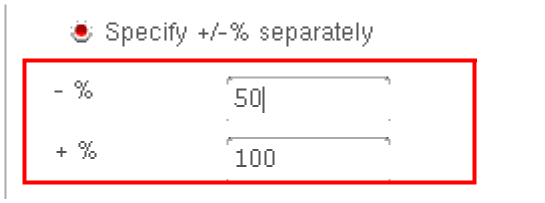
3. In this form, you can specify parameter ranges in one of the following two ways:

- ❑ **Parameter range as percentage of the design value:** This is the default way. In this case, the *Percentage of design value* option is selected and the parameter range is relative to the value of variable. If the variable is assigned any value in the Data View pane, that value is taken as a reference. If there is no value assigned to this variable in the Data View pane, the value for this variable is taken from the design.

  By default, you can specify a common percentage value for both the upper and the lower limit in the *+/- %* field. The lower limit is the value of a parameter in the design minus the specified % of the same design value. Similarly, the upper limit is the value of a parameter in the design added to the specified % of the value of parameter in the design.

  Note: The +/- % field can contain only an integer value between `1` and `100`, including these two values.

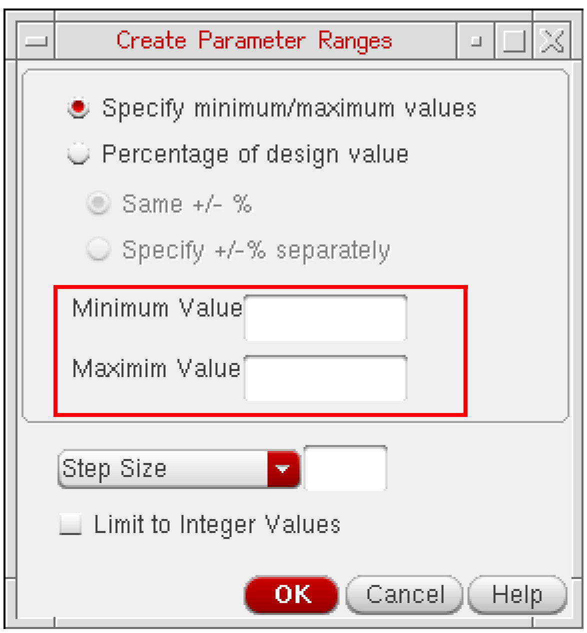
Alternatively, select the *Specify +/- % separately* option. When you select this option, the following fields appear in the form.



In these fields, you can specify separate % values for the upper and lower range limits.

Note: The *-%* field can contain an integer between `0` and `100`, including these two values, and the *+%* can be set to any non-negative integer.

- **Parameter range with a specific minimum and maximum value:** This option is used to specify specific minimum and maximum value for the parameter range. These values might not be relative to the value of a parameter in the design. To specify these values, select the *Specify minimum/maximum values* option. The *Minimum Value* and *Maximum Value* fields appear in the form.



Specify the minimum and maximum values of the parameter range in their respective fields.

Tip

To set the value of a parameter same as given in the design, right click on the parameter name and choose *Set to Design Value*.

4. Next, you can specify how to create intermediate values within the range. For this, you have the following two options in the drop-down list displayed in the lower section of the form:

   - *Step Size*: Use this option to specify a step size between two steps. The tool automatically creates steps between the two outer limits of the range at a gap of the given step size. In this case, ranges of all the design parameters will have the same step size, but the number of step values will depend on the minimum and maximum values.

   - *Number of Step Values*: Use this option to specify the number of step values to be created between the two range limits. The tool automatically calculates the required step size and creates the intermediate values. In this case, the number of step values will remain same for all the design parameters, but the step size will vary.

5. Select the *Limit to Integer Values* option to specify that the step values should be only integer numbers. This would be required for parameters that cannot have decimal values, such as m-factor and number of fingers. When you select this option, parameter range is not created if the step values contain decimal values.

   Note: An alternate way to limit the creation of range for parameters defining m-factor and number of fingers to only integer values is to add the names of such parameters in the `layoutXL.mFactorNames` and `layoutXL.lxFingeringNames` environment variables. ADE XL automatically sets only integer ranges for the parameters listed in these variables.

6. Click *OK* to close the form.

   ADE XL creates the parameter ranges for all the parameters and updates the Data View and the Variables and Parameters assistants.

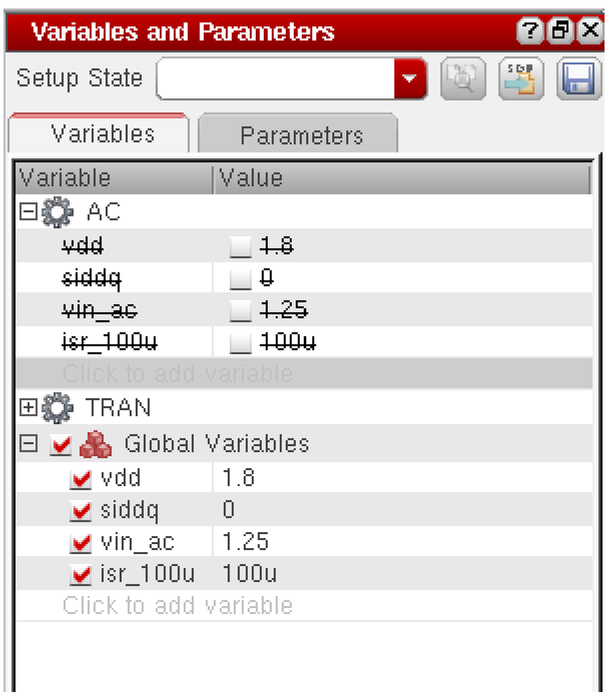
   When the Variables and Parameters assistant is open in the schematic view and you select an instance to create a parameter, you can right-click on the instance in the top area of the assistant and use the *Create Parameter Range* command to create a parameter as well as to apply a range of values for it.

4

Working with Global Variables

Global variables can be used to override the design variable values in the tests without having to modify the value in the test itself. In addition, global variables can be used to sweep the design variable values. This allows you to change the value of global variables so that the simulator will use the value you specified and not the value from the schematic.

By default, the design variables defined for your tests are added as global variables in the *Global Variables* tree on the Data View assistant pane and the Variables tab of the Variables and Parameters assistant pane. For example, in the following figure, the design variables specified for the test named `myFirstTest` are automatically added as global variables.



The design variables are displayed with a ~~strikethrough~~ because the settings for a global variable override the settings of design variables that have the same name as the global variable.

Note: You can disable the default automatic copy of design variables to global variables by setting the autoPromoteVarsToGlobal environment variable to `nil`. However, after disabling automatic copy of design variables, you can add a global variable with the same name as one of the design variable. For more details, see Creating a Global Variable. When you do this, the design variable for the test is automatically displayed with a ~~strikethrough~~ and the global variable is applied to the test. However, you can choose to disable the global variable for a test. For more details, see Disabling Global Variables for Specific Tests on page 170.

For more information, see the following:

- Creating a Global Variable on page 163
- Loading a Set of Global Variables from a File on page 165
- Importing Sweep Variables as Global Variables on page 166
- Saving Global Variables to an ADE State on page 166
- Importing Global Variables from a Saved ADE State on page 167
- Enabling and Disabling Global Variables for All Tests on page 168
- Disabling Global Variables for Specific Tests on page 170
- Specifying an Instance Parameter as a Sweep Parameter on page 178
- Working with Parametric Sets on page 182
- Working with Parametric Sets on page 182
- Enabling and Disabling Parameters on page 189
- Adding or Changing a Parameter Specification on page 191
- Deleting a Parameter Specification on page 201
- Creating Matched Device Parameters on page 201
- Creating Ratio-Matched Device Parameters on page 204
- Creating a Combinatorial Expression on page 208
- Toggling the View on the Variables tab of the Variables and Parameters Assistant Pane on page 209
- Sorting Parameters by Properties and Objects on page 210
- Disabling Callbacks on Swept Device Parameters on page 212

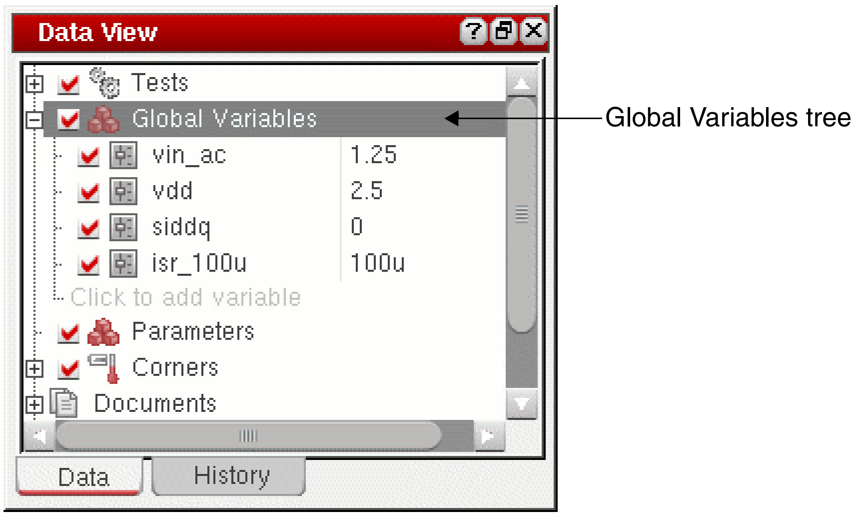
See also

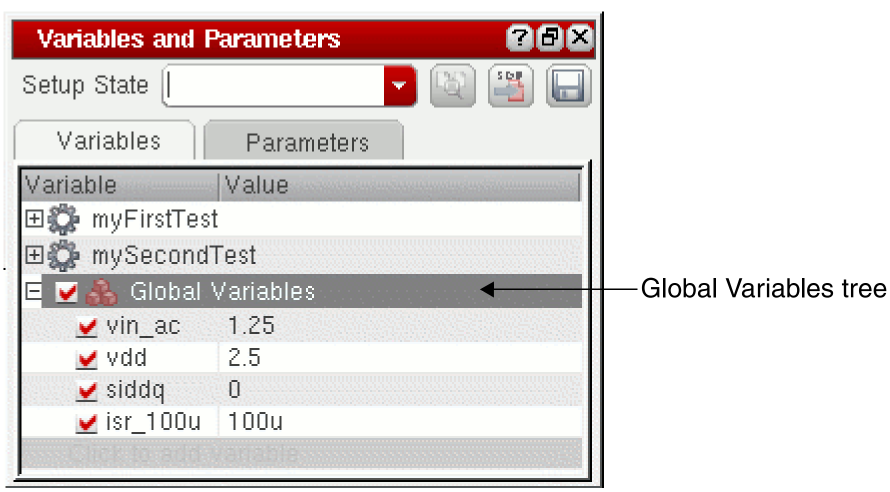
- Varying the Model File and Section during Simulation on page 359
- Variables and Parameters Assistant Right-Click Menus on page 214

Creating a Global Variable

To create a new global variable, do the following:

1. On the Data View assistant pane or the Variables tab of the Variables and Parameters assistant pane, right-click on *Global Variables* and choose *Add Global Variable*.

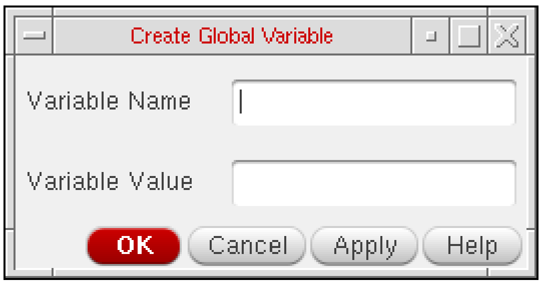




Tip

Alternatively, click where it says *Click to add variable* in an expanded *Global Variables* tree on the Data View pane or the Variables tab of the Variables and Parameters pane.

The Create Global Variable form appears.



2. In the *Variable Name* field, type a name for your global variable.

   The name must begin with a letter or number and can contain only letters and numbers.

3. In the *Variable Value* field, type the value for your global variable.

4. Click *OK*.

   Your global variable appears in the *Global Variables* tree on the Date View assistant pane and the Variables tab of the Variables and Parameters assistant pane. A global variable setting takes precedence over all other settings for variables that have the same name (such as a design variable with the same name).

5. To define sweep values for this global variable, do one of the following:

   ❑ Double-click on the *Value* field next to the global variable and modify the value.

   ❑ Double-click on the *Value* field next to the global variable, then click the browse button to define a parameter specification. For more information, see Adding or Changing a Parameter Specification.

If you need to save some information about a global variable, you can add notes for it. For this, right-click on the global variable on the Data View pane and click *Notes* to open the Add/Edit Notes form. Add notes in the *Notes* field and click *OK*. These notes are displayed in the tooltip for the variable and saved in the setup database.

For related information, see Adding Notes to a Test.

Loading a Set of Global Variables from a File

You can load a set of variables that appears in the *Global Variables* tree on the Date View assistant pane and the Variables tab of the Variables and Parameters assistant pane whenever you add a new test. You can define a distinct set of default variables for each library. You can also define one set of default variables not associated with any library.

When you open an ADE XL setup, the program loads the set of default variables associated with the same library as the setup (if it exists). After that, it loads the set of default variables not associated with any library (if it exists).

To create and load a set of global variables, do the following:

1. Create a file containing exactly one `axlSetDefaultVariables` SKILL function call for each set of global variables you want to define. (For example, `adexlGlobals.il`.)
   - ❑ For global variables particular to a library, one call to the function for each library:

     `axlSetDefaultVariables( '(global1 value1 global2 value2) "libName" )`

     For example:

     `axlSetDefaultVariables( '(_n_len 1u _sim_time 100n) "demoLib" )`
   - ❑ For global variables not for a particular library, exactly one call to the function:

     `axlSetDefaultVariables( '(global1 value1 global2 value2) )`

     For example:

     `axlSetDefaultVariables( '(_n_len 1u _sim_time 100n) )`
2. You can load the file into the CIW using the `load` command.

   For example:

   `load("adexlGlobals.il")`

Tip

You can put this load command in your `.cdsinit` file so that the program loads it every time you run the software.

If your file contains more than one call for a library or more than one call that does not name a library, the last call takes precedence such that the program loads only those global variables that you defined in each of the last of these calls.

Importing Sweep Variables as Global Variables

You can save the sweep settings you specified in the Parametric Analysis tool in the Virtuoso Analog Design Environment L (ADE L) in a file. You can then import the sweep variables in the file as global variables. For more information about parametric analysis in ADE L, see the *Virtuoso Analog Design Environment L User Guide*.

To import sweep variables as global variables, do the following:

1. On the Data View pane or the Variables tab of the Variables and Parameters pane, right-click a global variable and choose *Import Sweep*.

   The Choose Parametric State File to Import form appears.

2. Select the file containing the sweep settings you saved in the Parametric Analysis tool in ADE L and click *Open*.

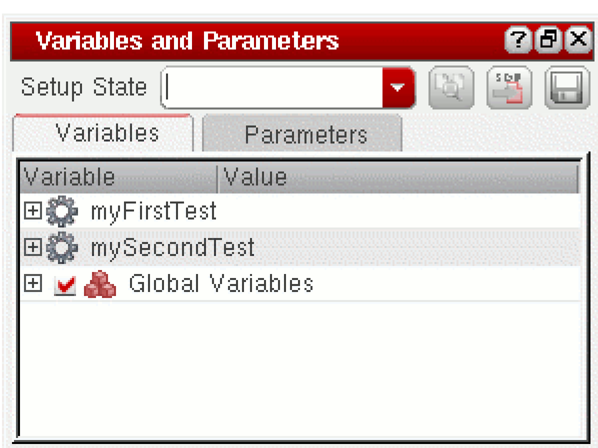
   The sweep variables in the file are displayed as global variables in the *Global Variables* tree on the Date View assistant pane and the Variables tab of the Variables and Parameters assistant pane.

Saving Global Variables to an ADE State

You can save the global variables and parameters from the Variables and Parameters assistant to an ADE state.

To save the variables and parameters, do the following:

1. After setting the variables and parameters and their values in the Variables and Parameters assistant, specify a name for the new ADE state in the *Setup State* field, as shown below.



2. Click *Save Variables and Parameters* ( ) to save the setup state.

Importing Global Variables from a Saved ADE State

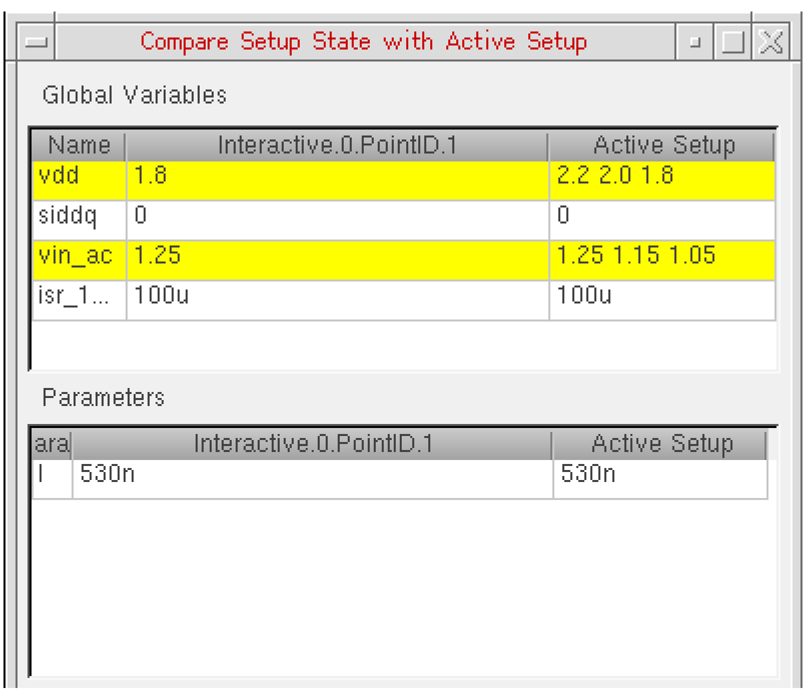
To import global variables and parameters along with their setup information from a saved state by using the Variables and Parameters assistant, do the following:

1. From the *Setup State* list, select name of the state from which you want to import the details.
2. Click *Load Variables and Parameters* ( ).

All the variables and parameters are loaded from the given ADE state and displayed in their respective tabs. Note that all the global variables and parameters are overwritten.

Comparing Variables and Parameters of a Setup State with the Active State

Before loading the global variables from a saved state, you can also choose to compare the variables and parameters with the currently active state. To do this, select name of a state from the *Setup State* list and click *Compare Setup State with Active Setup* ( ). The comparison is displayed in the **Compare Setup State with Active Setup** form, as shown below.



The variables and parameters that have different values in the two setups are highlighted in yellow.

Click *OK* to close the form.

You can also import the global variables and parameters by using the **Loading State** form. For more details, see "Loading State Information" on page 92.

Enabling and Disabling Global Variables for All Tests

You can enable or disable global variables before starting your simulation. This allows you to specify the set of global variables you want to be swept for a particular simulation.

To enable a global variable for all tests,

➤ In the *Global Variables* tree on the Data View pane or the Variables tab of the Variables and Parameters pane, select the check box to the left of the variable.

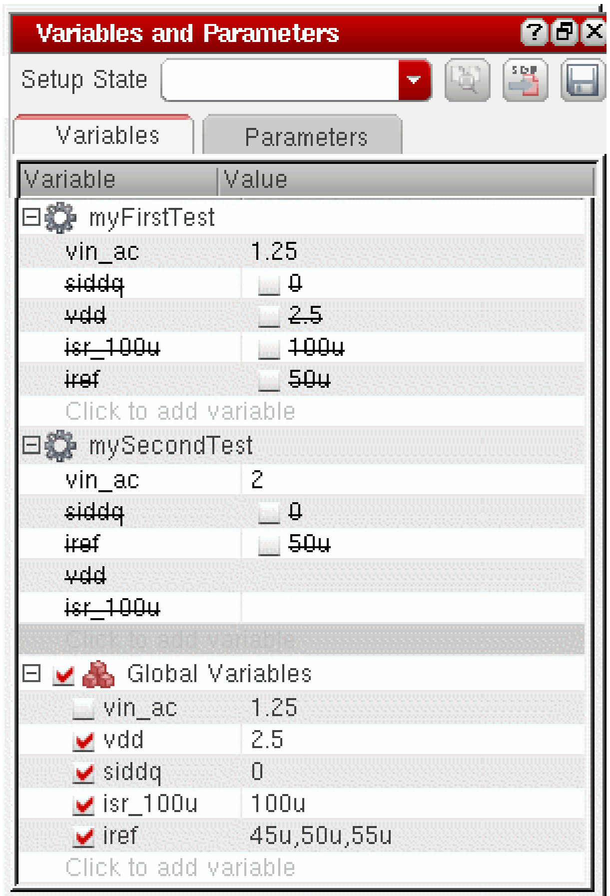
Alternatively, right-click on the variable and choose *Enable/Disable* from the pop-up menu.

To disable a global variable for all tests,

➤ In the *Global Variables* tree on the Data View pane or the Variables tab of the Variables and Parameters pane, clear the check box to the left of the variable.

Alternatively, right-click on the variable and choose *Enable/Disable* from the pop-up menu.

When a global variable is disabled, its value will not be used for simulation but its definition is retained in the *Global Variables* tree on the Data View pane and the Variables tab of the Variables and Parameters pane. The value of the design variable (specified for a test) that has the same name as the global variable will be used for the simulation. The ~~strikethrough~~ no longer appears for the design variable value in the test tree. For example, in the following figure, the value of design variable `vin_ac` specified for the tests named `myFirstTest` and `mySecondTest` will be used because the global variable `vin_ac` is disabled.



To toggle the enabling or disabling of more than one global variable,

➤ In the *Global Variables* tree on the Data View pane or the Variables tab of the Variables and Parameters pane, select the variables, right-click and choose *Enable/Disable* from the pop-up menu.

The enabled variables are disabled and the disabled variables are enabled.

When all global variables are disabled, global variable values will not be used for simulation but their definition is retained in the *Global Variables* tree on the Data View pane and the Variables tab of the Variables and Parameters pane. The value of design variables (specified for tests) that have the same name as the global variable will be used for the simulation.

Disabling Global Variables for Specific Tests

By default, global variables are swept over all the tests that are enabled in the Data View pane. You can disable a global variable for specific tests if you do not want to sweep the global variable over those tests. If a global variable is disabled for a test, the value of the design variable (specified for the test) that has the same name as the global variable will be used for simulation and for that you can specify only a single value.

You can disable a design variable for a test in any of the following ways:

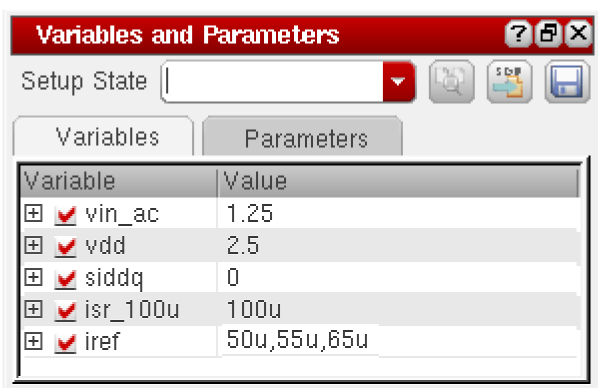
- Using the Variables and Parameters assistant
- Using the Data View assistant

Disabling a Global Variable for a Test using the Variables and Parameters Assistant

To disable global variables using the Variables and Parameters assistant pane.

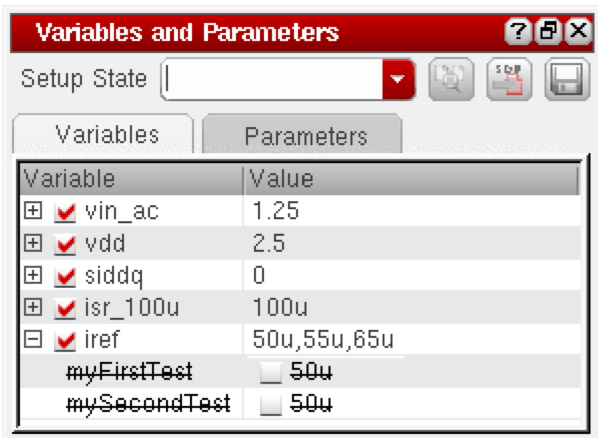
1. Click the Variables tab of the Variables and Parameters pane.

   The *Variables* tab displays the list of global variables as shown below:



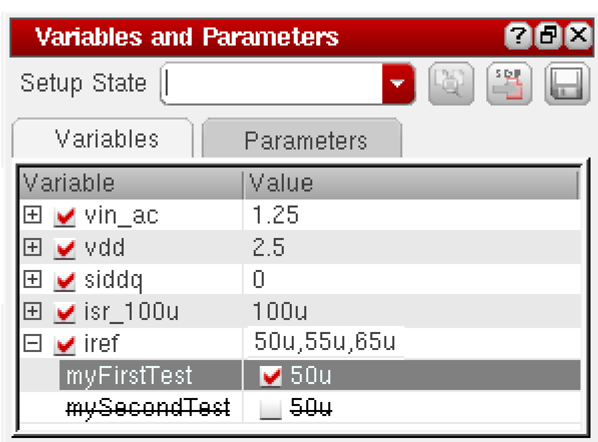
2. Click the plus sign to the left of the global variable name to view the names of tests over which the global variable will be swept during simulation.

   For example, the following figure indicates that the values `45u`, `50u` and `55u` specified for global variable `iref` will be swept over the tests named `myFirstTest and mySecondTest`. In this example, the `myFirstTest` and `mySecondTest` tests have a design variable named `iref` with the value `50u`. The test names and values are displayed with a ~~strikethrough~~ because the global variable `iref` overrides the design variable `iref`.



3. In the *Value* column, select the check box next to the name of the test for which you want to disable the global variable.

   The global variable is disabled for the test. For example, in the following figure, the global variable `iref` is disabled for the test named `myFirstTest`. Because of this, the value `50u` will be used for the test during simulation. The global variable will be swept only for the test named `mySecondTest`.

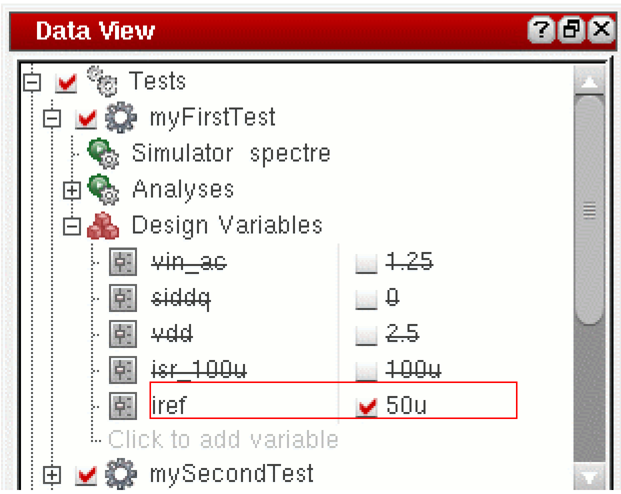


Also see:

- Hiding Overridden Design Variables for a Test on page 136
- How Results are Displayed When A Global Variable is Disabled for a Test? on page 174

Disabling a Global Variable for a Test using the Data View Assistant

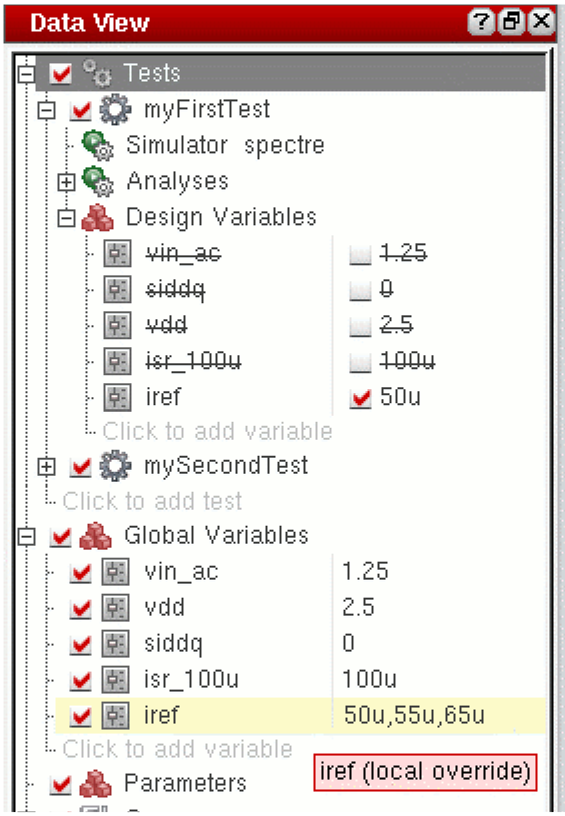
You can disable a global variable for a test using the Data View assistant also, as shown in the following figure:



Note that the global variable `iref` is using the value specified for the test `myFirstTest` instead of using the sweep points specified for the global variable.

The settings made using the Data View assistant are also reflected in the Variables and Parameters assistant and vice-versa.

In the Data View pane, if you place the mouse pointer on the global variable that is disabled for a test, a tooltip is displayed containing the text `(local override)`, as shown in the following figure:



Also see:

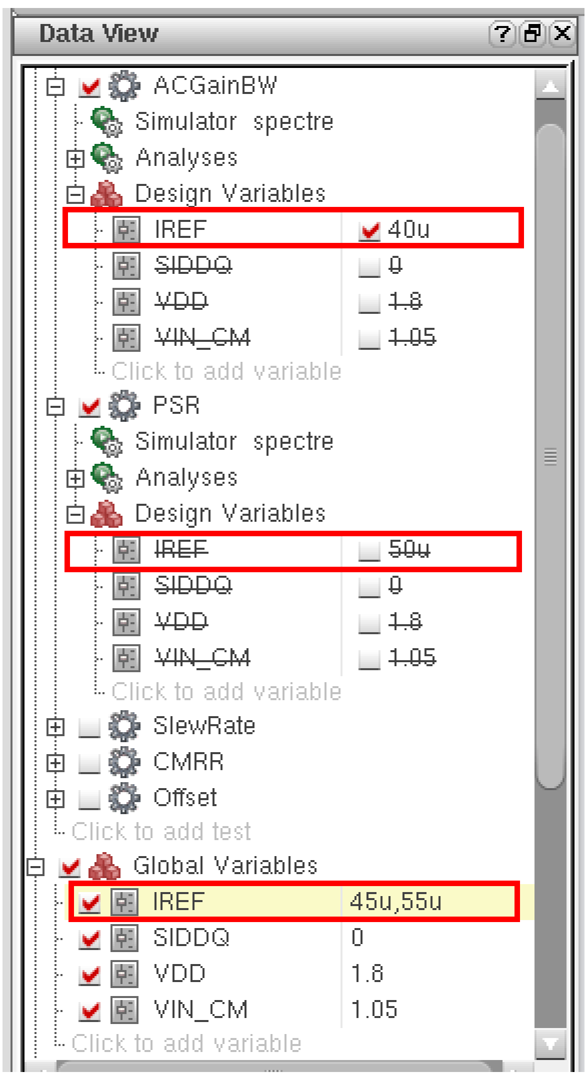
- Hiding Overridden Design Variables for a Test on page 136
- How Results are Displayed When A Global Variable is Disabled for a Test? on page 174

How Results are Displayed When A Global Variable is Disabled for a Test?

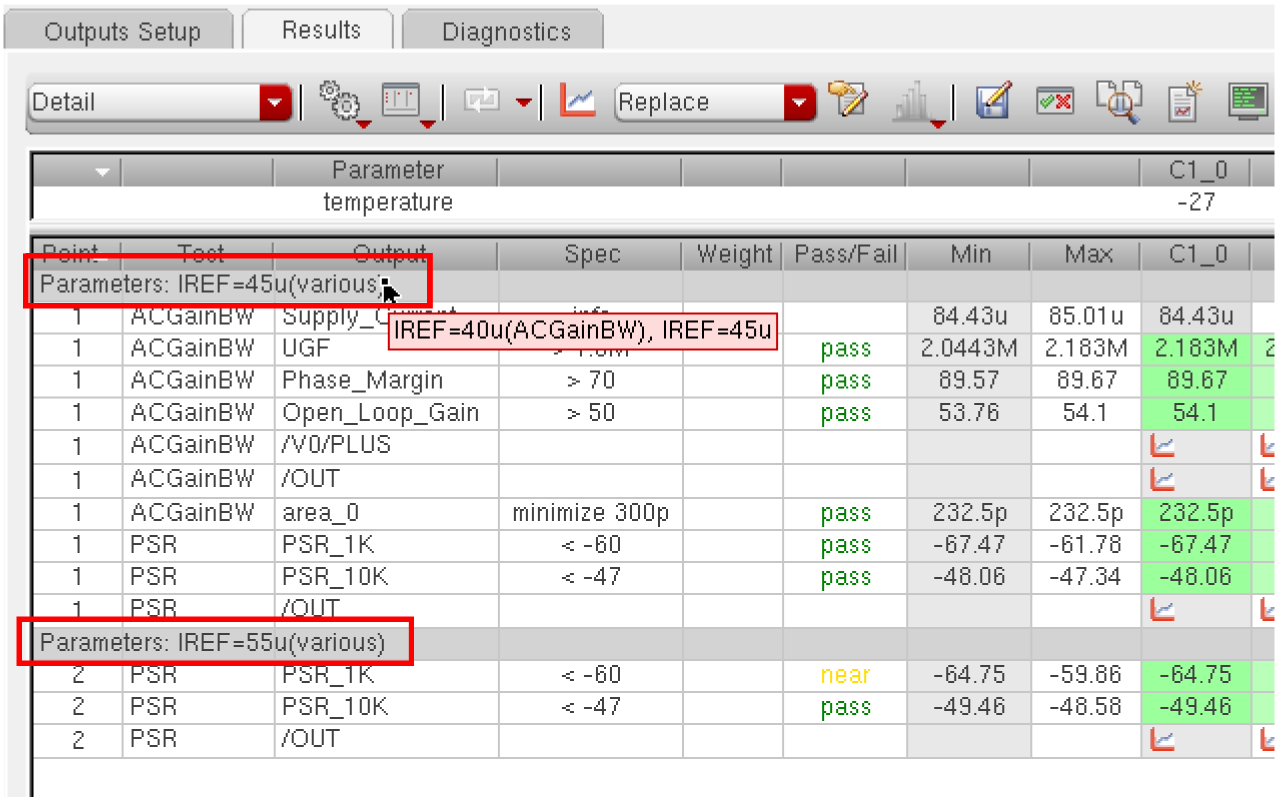
The *Results* tab displays the simulation results parameter-wise. Each unique combination of the parameter values is displayed on a parameter header for a design point.

If a global sweep variable is disabled for a test, its simulation is run only for the local value of that variable, which is the value set for its design variable with the same name. For the other tests, the simulation is run for each sweep value specified for the global variable.

Consider the following example where the ADE XL setup has two tests, `ACGainBW` and `PSR`. The `IREF` global variable has two sweep values, `45u` and `55u`, and it is disabled for the test `ACGainBW`, which uses a local value `40u`. The *Data View* assistant given below shows how the variables are set.



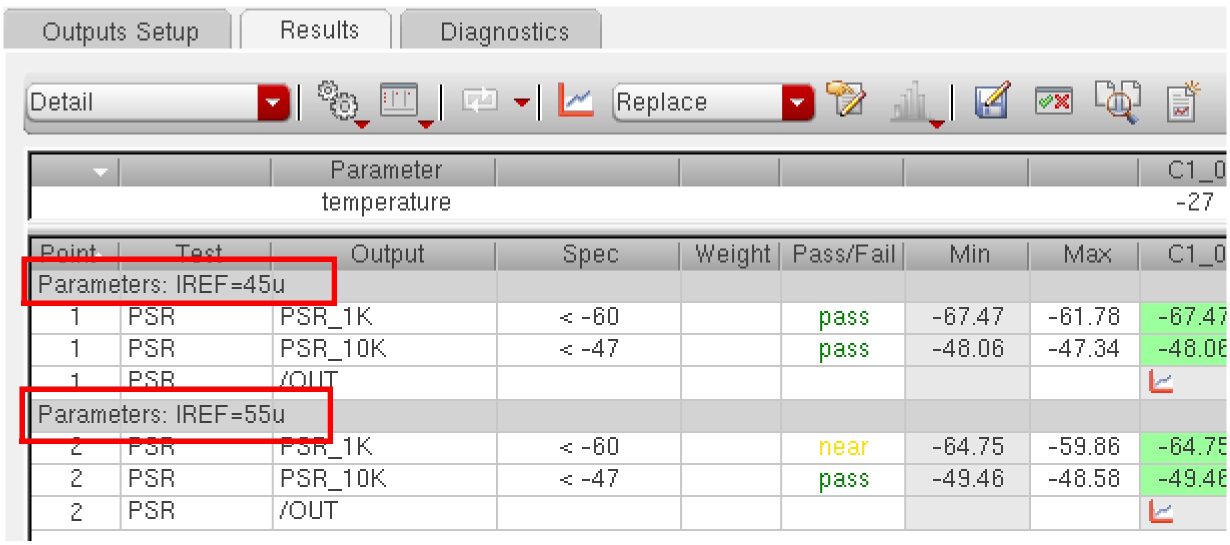
In this case, the results are displayed as shown in the figure below.



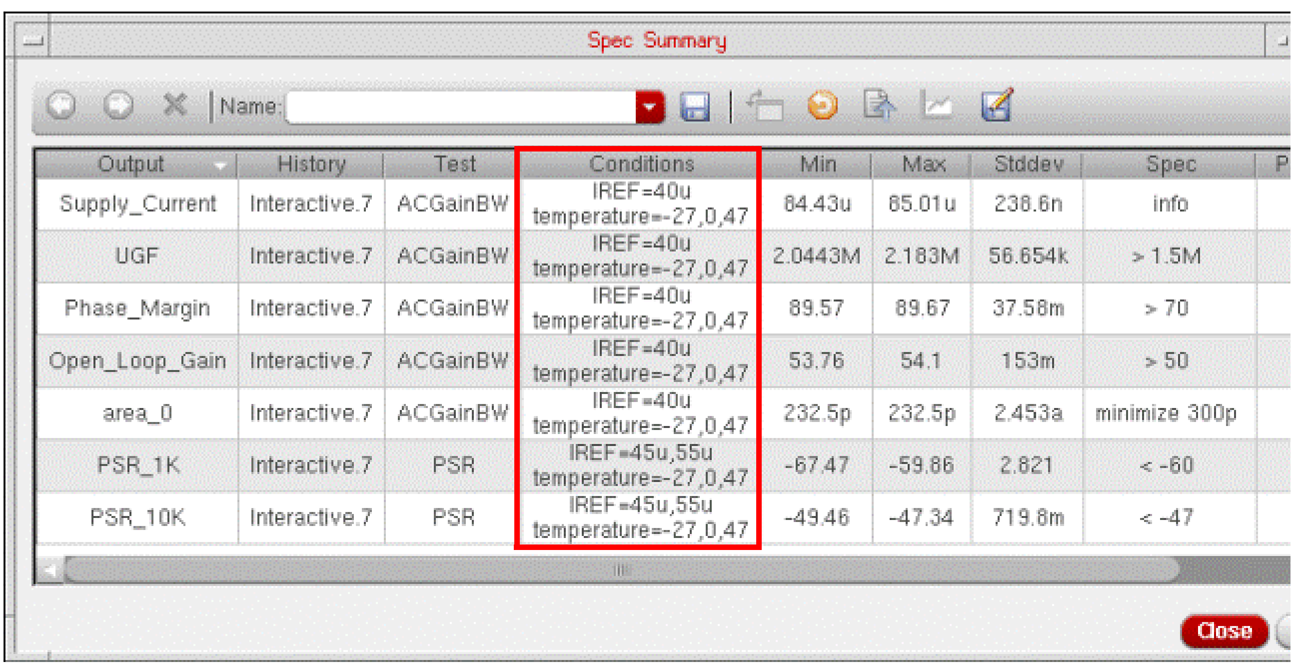
Note the following points:

- Two design points have been generated. For the first point, the value of `IREF` used by `ACGainBW` is `40u` (local), whereas the value used by `PSR` is `45u` (global).
- For the second design point, the simulation is run only for `PSR` with the value of `IREF` set to `55u` (global). The rows for `ACGainBW` are hidden for this design point.
- When a variable is disabled for a particular test, the *Parameters* headers (the gray rows marking the design points) contain the text `(various)`. When you move the pointer on a header, a tooltip displays the values used for the design point.
- If you apply a filter to show only a selected set of tests, the parameter header rows are modified accordingly to show the variable values. For example, in the *Results* tab shown above, if you apply a filter to hide the results for `ACGainBW`, the parameter header rows

are updated automatically to show the value of `IREF` being used by `PSR` only. The text `(various)` is also removed, as shown in the figure below.



- The spec summary and spec comparison generated using the results of tests that have disabled global variables show all the values used by different outputs. For example, the spec summary for the example given above shows all the variable conditions used for each output.



For details on how to generate Spec Summary and Spec Comparison, refer to Working with Specifications on page 701.

Updating Global Variable Values with Design Variable Values

To update the value of a global variable with that of a corresponding design variable, do the following:

➡ In an expanded test tree on the Data View pane or on the Variables tab of the Variables and Parameters pane, right-click a design variable and choose *Create/Update Global*.

The value of the global variable that has the same name as the design variable is updated.

Updating Design Variable Values with Global Variable Values

To update the value of a test's design variable with that of the corresponding global variable, do the following:

➡ In the *Global Variables* tree on the Data View pane or the Variables tab of the Variables and Parameters pane, right-click a variable, choose *Backannotate to Test* and do one of the following:

Choose	To
All	Update the values of design variables that have the same name as the global variable in all the tests.
Testname	Update the value of a design variable that has the same name as the global variable in the selected test. **Note:** If a design variable with the same name does not exist for the test, the design variable is automatically added for the test.

Note: You cannot update a design variable whose corresponding global variable has a sweep value.

Specifying an Instance Parameter as a Sweep Parameter

You can specify a device instance parameter as a sweep parameter. The settings for a sweep parameter override the settings of an instance parameter that has the same name as the sweep parameter.

To specify an instance parameter as a sweep parameter, do the following:

1. Make sure you are viewing the design schematic associated with your test.

   For more information, see Opening a Design Schematic on page 90.

2. Choose *Window – Assistant – Variables and Parameters* to display the Variables and Parameters assistant pane.

3. Click the Parameters tab on the Variables and Parameters assistant pane.

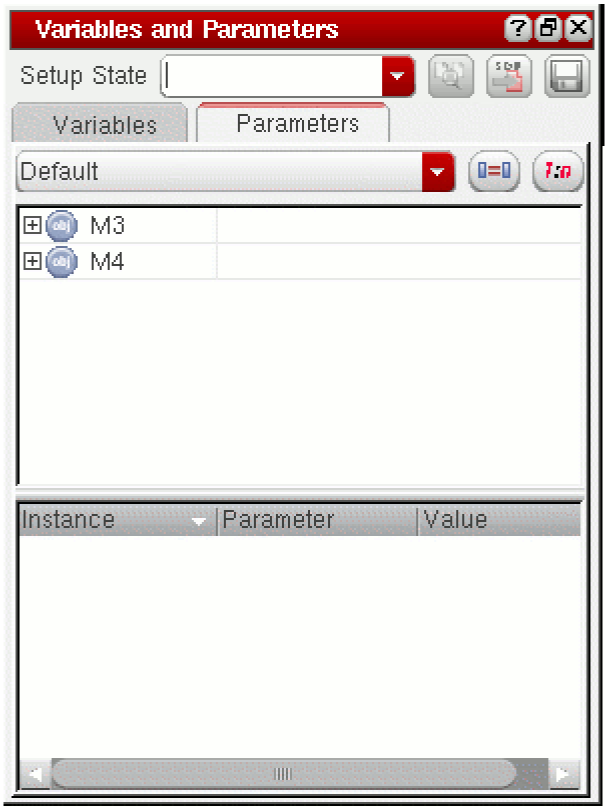
4. On the schematic, select the instances whose parameters you want to specify as sweep parameters.

   You can select more than one instance by shift-clicking each instance one by one or by dragging a selection box around a set of instances.

Tip

Use the Navigator assistant pane to quickly select the instance.

The selected instances are displayed on the upper half of the Parameters tab of the Variables and Parameters pane.

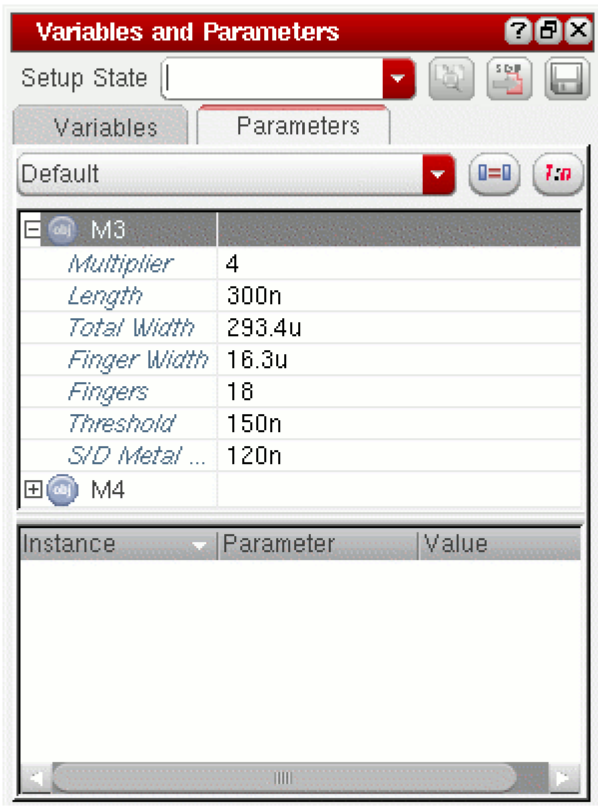


Tip

Place the mouse pointer on the instance name to view the name of the library, cell, and view containing the instance in a pop-up.

5. Click the plus sign to the left of the instance name to view the parameters for the instance.

The parameters associated with the instance appear in the expanded tree beneath the instance name. The value of each parameter appears in the second column.



Tip

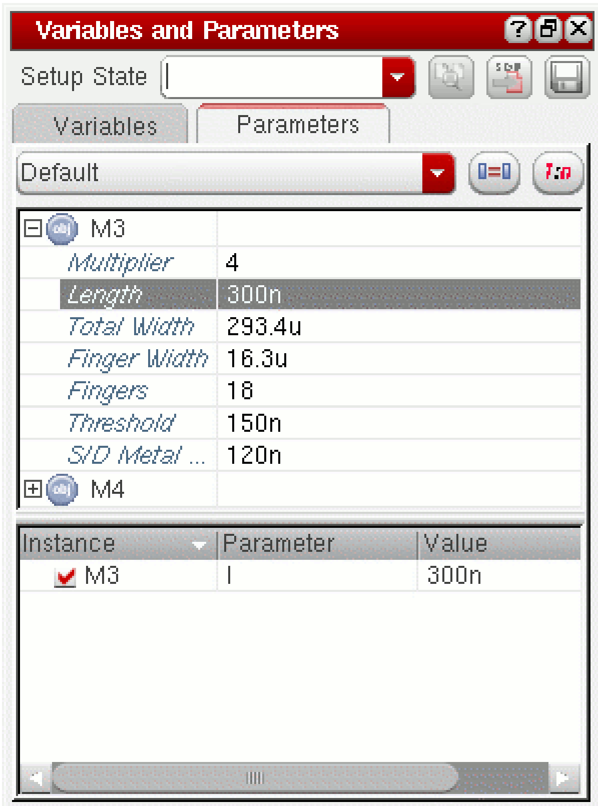
To view only the parameters in the Parameters tab, see Sorting Parameters by Properties and Objects on page 210.

See also

- Filtering Device Instance Parameters on page 152
- Creating Custom Device Filters on page 153
- Sorting Parameters by Properties and Objects on page 210

6. Right-click the instance parameter you want to use as a sweep parameter and choose *Create Parameter*.

The instance parameter and its value appear on the lower half of the Parameters tab on the Variables and Parameters assistant pane. See also "Toggling the View on the Variables tab of the Variables and Parameters Assistant Pane" on page 209.



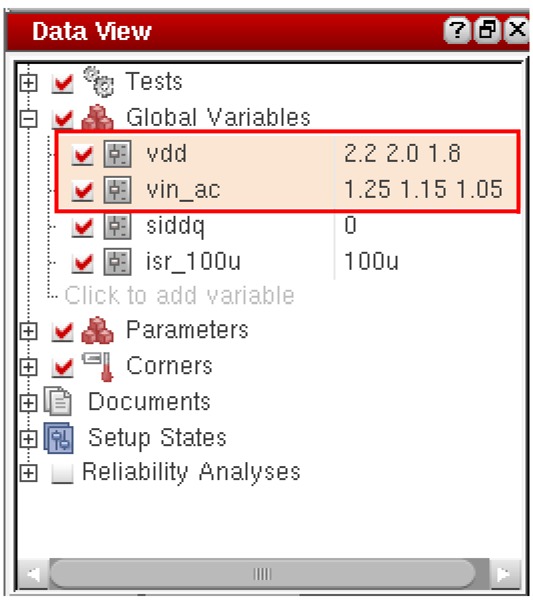
7. To define sweep values for this parameter, do one of the following:

- ❑ Double-click the value of the parameter on the lower half of the Parameters tab and specify a sweep value.
- ❑ Double-click the value of the parameter on the lower half of the Parameters tab, then click the browse button to define a parameter specification. For more information, see "Adding or Changing a Parameter Specification" on page 191.

Working with Parametric Sets

By default, ADE XL creates all possible sweep combinations by pairing each value of a variable or parameter with all values of the other variables or parameters. With parametric sets, a selected set of sweep combinations are created by picking values from the same ordinal position for all the variables or parameters in that parametric set. This reduces the number of design points, thus reducing the number of simulations.

Consider an example where two variables, `vdd` and `vin_ac`, are grouped together to create a parametric set as shown below.



In this case, ADE XL creates the following three sweep value sets:

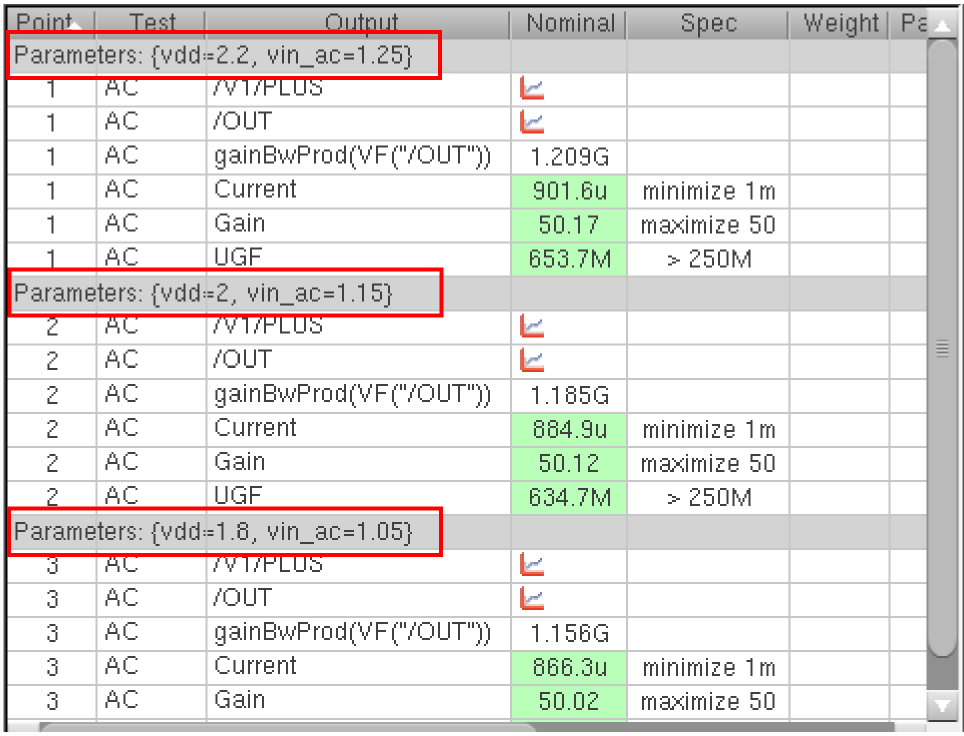
vdd: `2.2`, *vin_ac*: `1.25`

vdd: `2.0`, *vin_ac*: `1.15`

vdd: `1.8`, *vin_ac*: `1.05`

Note that the sweep value sets are created by picking values from the same ordinal position for all the variables or parameters in a parametric set.

You can see that while running simulations, ADE XL created three design points, as shown below.



Point	Test	Output	Nominal	Spec	Weight	Pa
Parameters: {vdd=2.2, vin_ac=1.25}						
1	AC	/V1/PLUS				
1	AC	/OUT				
1	AC	gainBwProd(VF("/OUT"))	1.209G			
1	AC	Current	901.6u	minimize 1m		
1	AC	Gain	50.17	maximize 50		
1	AC	UGF	653.7M	> 250M		
Parameters: {vdd=2, vin_ac=1.15}						
2	AC	/V1/PLUS				
2	AC	/OUT				
2	AC	gainBwProd(VF("/OUT"))	1.185G			
2	AC	Current	884.9u	minimize 1m		
2	AC	Gain	50.12	maximize 50		
2	AC	UGF	634.7M	> 250M		
Parameters: {vdd=1.8, vin_ac=1.05}						
3	AC	/V1/PLUS				
3	AC	/OUT				
3	AC	gainBwProd(VF("/OUT"))	1.156G			
3	AC	Current	866.3u	minimize 1m		
3	AC	Gain	50.02	maximize 50		

As compared to this, if the parametric sets are not created, ADE XL creates nine design points, where each value of `vdd` is paired with all three values of `vin_ac`.

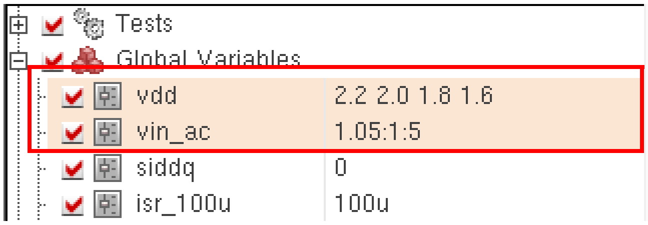
Note: For each design point, the parametric sets are indicated in the Results tab by using the {} brackets.

You can create parametric sets when you need to run simulations for a specific set of sweep values for different parameters. This helps in saving the simulation run time.

Creating Parametric Sets

To create a parametric set, do the following in the Data View pane or the Variables and Parameters assistant:

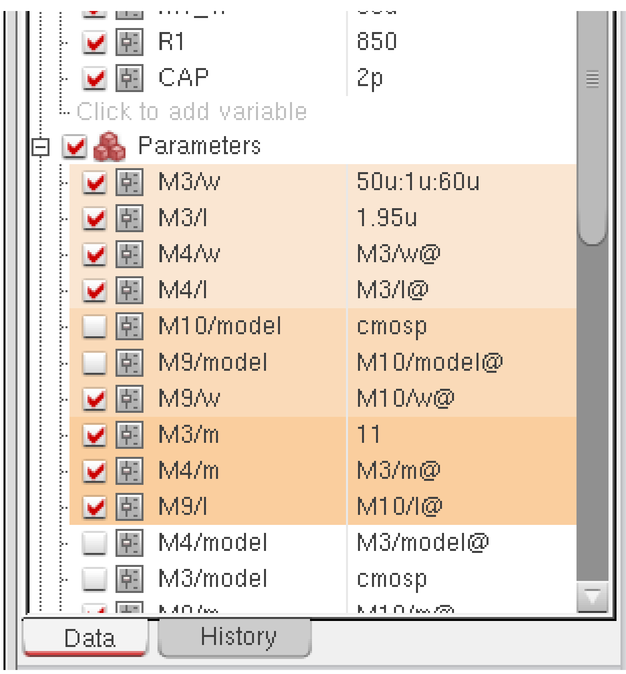
1. Hold down the *Ctrl* key and select two or more variables that you want to group together.
2. Right-click and choose *Group as Parametric Set*.
3. If required, specify sweep values for the variables. Note that for different variables, you can specify sweep values in different format. For example, you can specify values for variables as shown below.



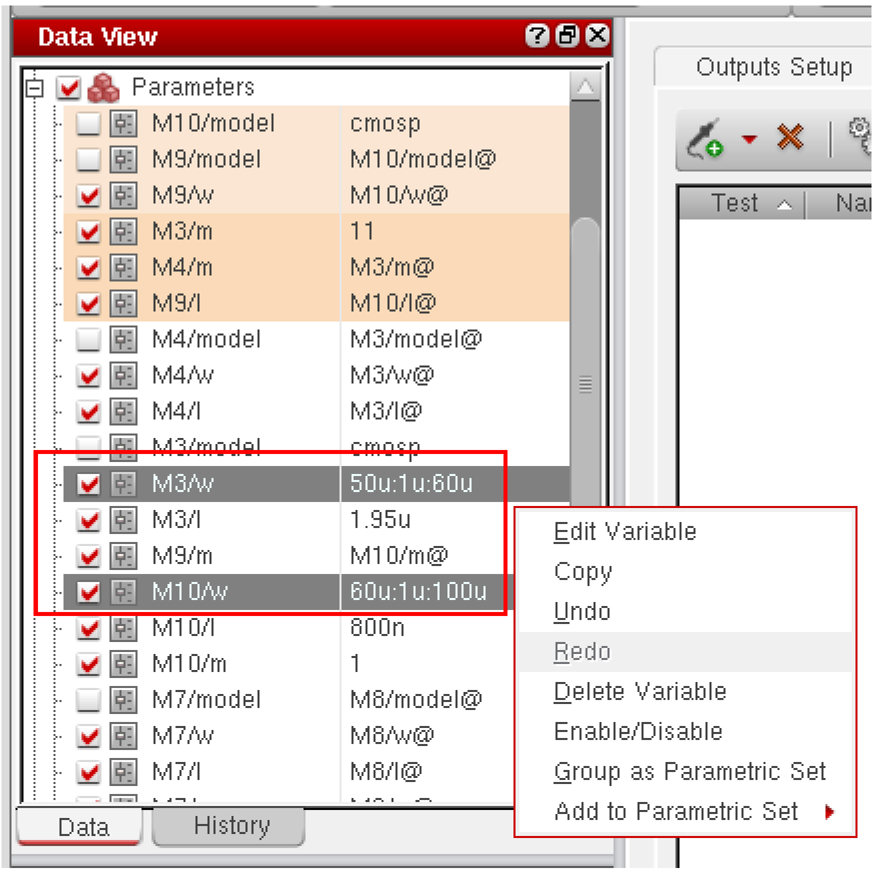
However, it is important that the number of sweep points is same for all the variables or parameters in a parametric set. If the number of sweep values are not equal, ADE XL flags an error during the simulation run.

Note the following changes in the Data View pane and the Variables and Parameters assistant:

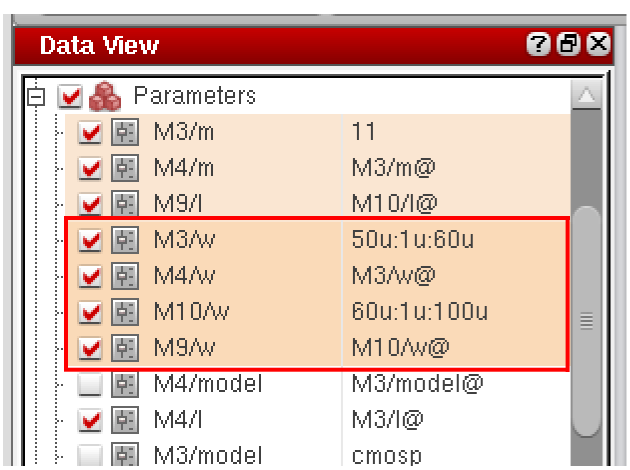
- The variables or parameters in a parametric set are bound as a group.
- When you create a parametric set, the list of variables is realigned to display all variables or parameters of a group together.
- Each parametric set is highlighted with a different color, as shown below.



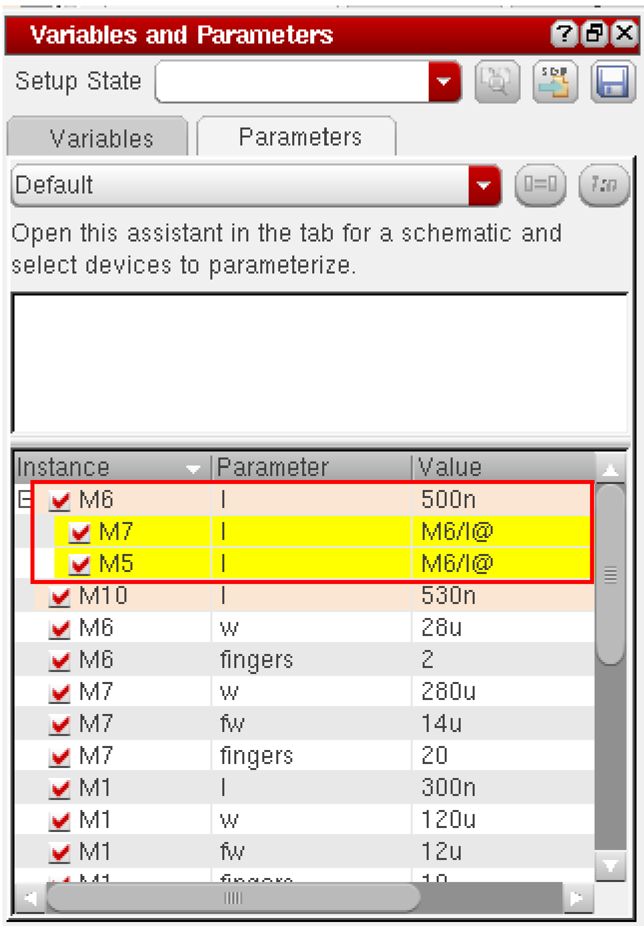
- When you group two more device parameters together, the tool automatically identifies matched parameters and includes then in the parametric set. Consider an example. From the *Parameters* list, select two variables, `M3/w` and `M10/w` and add them to a parametric set, as shown below.



A parametric set is formed and two more variables M4/w and M9/w are also added to that set. This is because m4/w and M9/w are matched with M3/w and M10/w, respectively.



- When you create a parametric set by using the Variables and Parameters assistant, the matched parameters in a parametric set are highlighted with yellow, as shown below.



- When you move the mouse over any variable in a parametric set, the tooltip shows that the variable is part of a parametric set.
- You cannot create a parametric set by including a mix of global variables and parameters.

Adding a Variable to a Parametric Set

To add a variable to an existing parametric set:

1. Right-click on a variable and choose *Add to Parametric Set*.
2. From the list of existing parametric sets shown in the submenu, select the name of the parametric set to which you want to add the parameter.

The variable is added to the specified parametric set. In addition, all the variables are realigned to display all variables or parameters of a group together.

Including a Dependent Variables in a Parametric Set

Dependent variables that have the same number of values as that of the other variables are supported in the parametric sets. The following examples show how dependent variables can be used in parametric sets:

Example 1:

A: 1, 2

B: 7, 8

C: A, B

D: 4, 5

In this example, a parametric set that includes variables C and D, is a valid set. This simulation will create 8 design points, as listed below.

Design point 1: A1=1, B1= 7, {C1=1, D1=4}

Design point 2: A1=1, B1= 7, {C1=7, D1=5}

Design point 3: A1=1, B1= 8, {C1=1, D1=4}

Design point 4: A1=1, B1= 8, {C1=8, D1=5}

Design point 5: A1=2, B1= 7, {C1=2, D1=4}

Design point 6: A1=2, B1= 7, {C1=7, D1=5}

Design point 7: A1=2, B1= 8, {C1=2, D1=4}

Design point 8: A1=2, B1= 8, {C1=8, D1=5}

Example 2:

A: 1, 2

B: 7

C: A, B

D: 4, 5

In this example, a parametric set that includes variables C and D, is a valid set. This simulation will create 4 design points, as listed below.s

Design point 1: A1=1, B1= 7, {C1=1, D1=4}

Design point 2: A1=1, B1= 7, {C1=7, D1=5}

Design point 3: A1=2, B1= 7, {C1=2, D1=4}

Design point 4: A1=2, B1= 7, {C1=7, D1=5}

Unsupported Dependent Variables

Dependent variables are not supported when:

- A dependent variable does not have the same number of values as that of the other variables

  For example:

  A: 1, 2

  B: A

  C: 3, 4

  In this example, A, B, and C cannot make a parametric set because B has only one value whereas A and C have two values.

- When there is cyclic dependency

  For example:

  A: 1, 2

  B: A, A

  C: 3, 4

  In this example, A, B, and C cannot make a parametric set because there is cyclic dependency between A and B.

Removing a Variable from a Parametric Set

To remove a variable from a parametric set:

➡ Right-click on that variable and choose *Remove from Parametric Set*.

The variable is removed from the parametric set and moved to its original location in the list.

Ungrouping Parametric Sets

To ungroup the parametric set:

➡ Right-click on any variable name in the group and choose *Ungroup Parametric Set*.

The group is dissolved and the variables are moved back to their original location in the Global Variables or Parameters list.

Enabling or Disabling Parametric Sets

You can disable or enable all parametric sets present in the Global Variables or Parameters list in the Data View pane.

To disable all parametric sets:

➡ In the Parameters list, right-click on *Parameters* and choose *Disable All Parametric Sets*.

After the parametric sets are disabled, the variables are moved back to their original location in the list.

Note: If the parametric sets are disabled, you can enable all the sets together. To do this, right-click on the list name, Global Variables or Parameters and choose *Enable All Parametric Sets*. All the parametric sets become visible again.

Enabling and Disabling Parameters

You can enable or disable parameters before starting your simulation. This allows you to specify the set of parameters you want to be used for a particular simulation.

To enable a parameter,

➤ In the *Parameters* tree on the Data View pane or the Parameters tab on the Variables and Parameters pane, select the check box to the left of the parameter.

Alternatively, right-click on the parameter and choose *Enable/Disable* from the pop-up menu.

To disable a parameter,

➤ In the *Parameters* tree on the Data View pane or the Parameters tab on the Variables and Parameters pane, clear the check box to the left of the parameter.

Alternatively, right-click on the parameter and choose *Enable/Disable* from the pop-up menu.

Note: When a parameter is disabled, its value given on the Data View pane is not used for simulation, but its definition is retained in the *Parameters* tree on the Data View pane and the Parameters tab on the Variables and Parameters pane. The value specified for the CDF parameter on the instance will be used for simulation.

Tip

To enable or disable more than one parameter at a time, select the parameters, right-click and choose *Enable/Disable* from the pop-up menu.

To toggle the enabling or disabling of one or more parameters,

➤ On the Parameters tab of the Variables and Parameters pane, select one or more parameters, right-click and choose *Toggle Enable/Disable*.

The enabled parameters are disabled and the disabled parameters are enabled.

To disable all parameters, do any of the following:

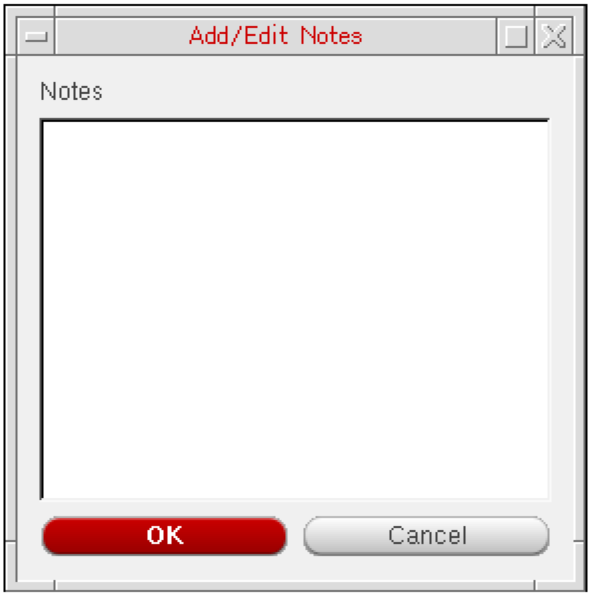
- On the Data View pane, clear the check box to the left of *Parameters*.
- Right-click on the Parameters tab of the Variables and Parameters pane and choose *Enable All/Disable All*.

Adding Notes for Parameters

If you need to save some additional reference information about a device parameters, you can add notes for it. For this, do the following:

1. On the *Data* tab of the Data View assistant pane, expand the *Parameters* tree, right-click the parameter for which you want to add notes, and choose *Notes*.

The Add/Edit Notes form is displayed.



2. In the *Notes* field, add notes for the device parameter.

   Note: By default, the notes field can accept only 512 characters. To change the default maximum characters limit, you can set the `maxNotesLength` environment variable.

3. Click *OK*.

The notes is added to the parameter. This is displayed in the tooltip for the parameter and is saved with the setup state.

For related information, see Adding Notes to a Test.

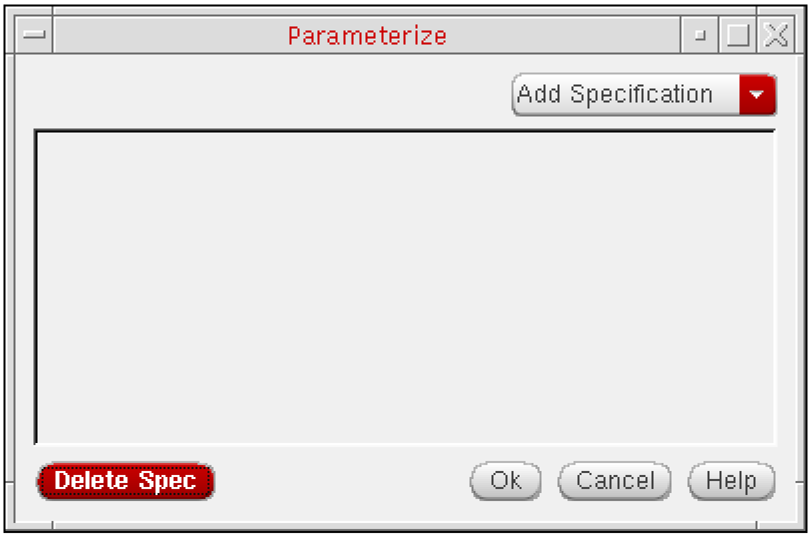
Adding or Changing a Parameter Specification

To add or change a parameter specification for a global variable you want to vary (sweep), do the following:

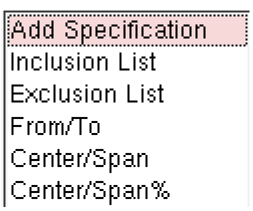
1. In the Global Variables tree on the Data View pane or the Variables tab of the Variables and Parameters pane, right-click the global variable for which you want to add or change the specification and choose *Edit Variable*.

   Alternatively, double-click on the *Value* field next to the global variable, then click the browse button.

The Parameterize form appears.



2. Click *Add Specification* and choose one of the following from the drop-down menu:



■ *Inclusion List* of values	You can specify a set of values through which you want to sweep your design variable.
■ *Exclusion List* of values	You can specify a set of values to exclude from your parametric sweep.
■ *From/To* range of values	You can specify a range of values through which you want to sweep your design variable and a method for stepping through that range.
■ *Center/Span* values	You can specify a center value and a span value for varying your design variable.
■ *Center/Span%* values	You can specify a center value and a percentage span value for varying your design variable.

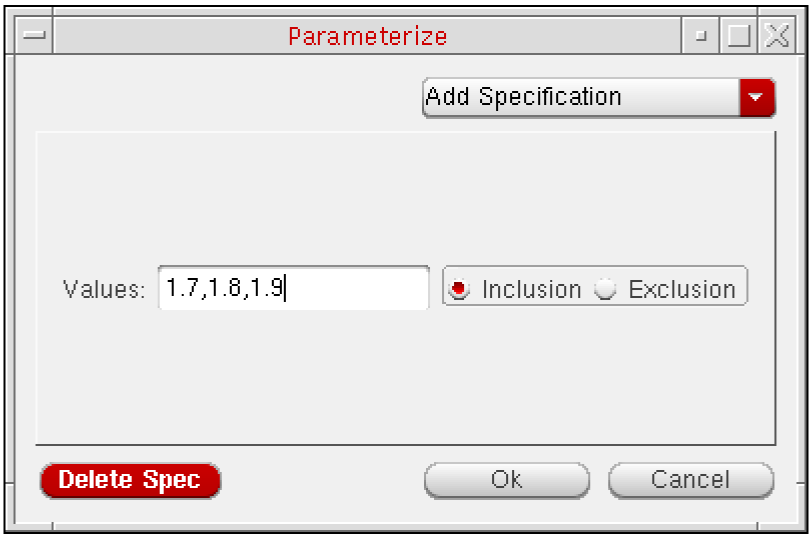
Specifying an Inclusion List of Values

To specify an inclusion list of values, do the following:

1. From the *Add Specification* drop-down menu, choose *Inclusion List*.

   A row appears on the Parameterize form for defining the specification type you selected. The *Inclusion* radio button is on.

2. In the *Values* field, type a list of values you want to include in your sweep.



   Note: You can separate the values in the list with either a comma or a space.

3. (Optional) Click *Add Specification* to add another parameter specification.

4. Click *OK*.

   The list of values appears in the *Value* column beginning and ending with the *{Inclusion List}* text string.

Note: You can change an inclusion list of values to an exclusion list (to exclude the list of values from the sweep) by selecting the *Exclusion* radio button in the row where your inclusion list is specified. See "Specifying an Exclusion List of Values" on page 194.

Specifying an Exclusion List of Values

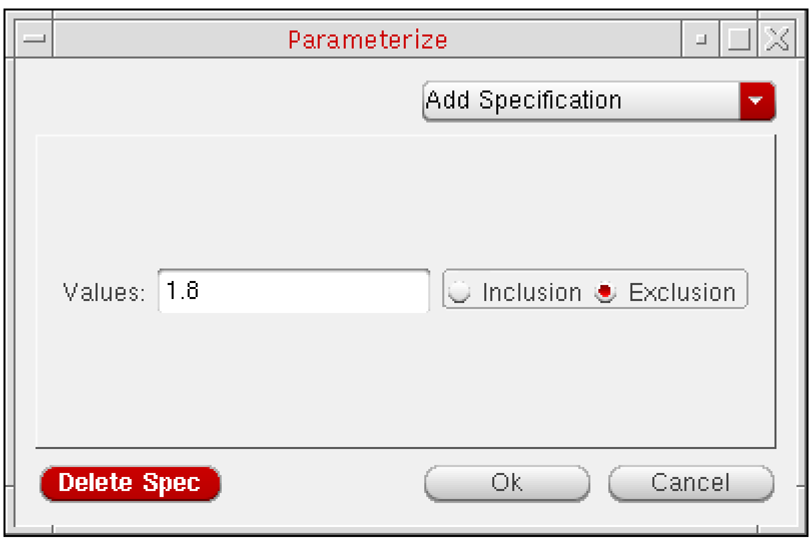
Tip

One application of the exclusion list is to define a *From/To* parameter specification (see "Specifying a Range of Values" on page 195) and exclude some values from the range by defining an exclusion list.

To specify an exclusion list of values, do the following:

1. From the *Add Specification* drop-down menu, choose *Exclusion List*.

   A row appears on the Parameterize form for defining the specification type you selected. The *Exclusion* radio button is on.



2. In the *Values* field, type a list of values you want to exclude from your sweep.

   Note: You can separate the values in the list with either a comma or a space.

3. (Optional) Click *Add Specification* to add another parameter specification.

4. Click *OK*.

   The list of values appears in the *Value* column beginning and ending with the *{Exclusion List}* text string.

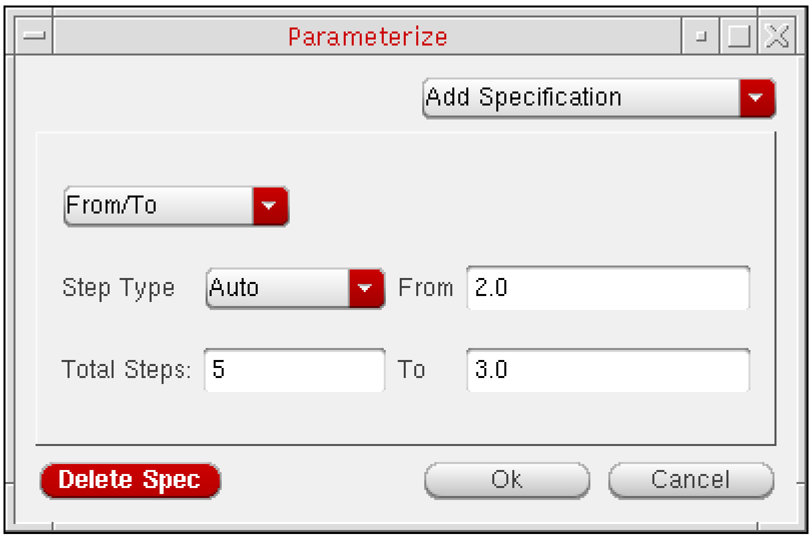
Note: You can change an exclusion list of values to an inclusion list (to include the list of values in the sweep) by selecting the *Inclusion* radio button in the row where your exclusion list is specified. See "Specifying an Inclusion List of Values" on page 193.

Specifying a Range of Values

To specify a range of values and how to sweep through those values, do the following:

1. From the *Add Specification* drop-down menu, choose *From/To*.

   A section appears on the Parameterize form for defining the specification type you selected.



2. In the *From* field, type the starting value for your range.
3. In the *To* field, type the ending value for your range.
4. From the *Step Type* drop-down menu, select how you want the simulator to step from one value to the next:

Linear	Linear steps from the *From* value to the *To* value
Decade	Decade steps from the *From* value to the *To* value
Octave	Octave steps from the *From* value to the *To* value
Logarithmic	Logarithmic steps from the *From* value to the *To* value
Times	Steps taken according to a specified multiplier from the *From* value to the *To* value

5. Depending on the *Step Type* you selected, type the remaining value for your parameter specification in the field that appears:

Step Type	**Field that appears**	**Value**
Auto	*Total Steps*	Total number of steps to take
Linear	*Step Size*	Size of steps to take when sweeping
Decade	*Steps/Decade*	How many steps to take per decade
Octave	*Steps/Octave*	How many steps to take per octave
Logarithmic	*Total Steps*	Total number of steps to take
Times	*Multiplier*	Multiplier for taking steps
		Note: Typing `10` for *Multiplier* is tantamount to selecting *Decade* and specifying `1` step per decade.

6. (Optional) Click *Add Specification* to add another parameter specification.

7. Click *OK*.

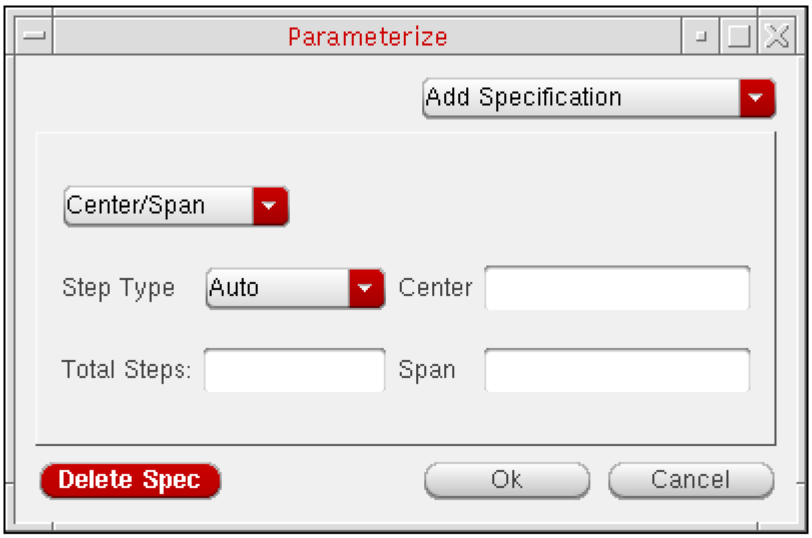
Note: You can change the specification type from *From/To* to *Center/Span* or *Center/Span%* by selecting a different item from the drop-down menu in the added section. See "Specifying Center and Span" on page 197 and "Specifying Center and Span as a Percentage" on page 199 for more information.

Specifying Center and Span

To specify a center value and a span value, do the following:

1. From the *Add Specification* drop-down menu, choose *Center/Span*.

   A section appears on the Parameterize form for defining the specification type you selected.



2. In the *Center* field, type the center value.
3. In the *Span* field, type a span value.

   The simulator can vary your parameter between *Center–Span* and *Center+Span* according to the *Step Type* you select (next).

4. From the *Step Type* drop-down menu, select how you want the simulator to vary the parameter value:

Linear	Linear steps from *Center–Span* to *Center+Span*
Decade	Decade steps from *Center–Span* to *Center+Span*
Octave	Octave steps from *Center–Span* to *Center+Span*
Logarithmic	Logarithmic steps from *Center–Span* to *Center+Span*
Times	Steps taken according to a specified multiplier from *Center–Span* to *Center+Span*

5. Depending on the *Step Type* you selected, type the remaining value for your parameter specification in the field that appears:

Step Type	**Field that appears**	**Value**
Auto	*Total Steps*	Total number of steps to take
Linear	*Step Size*	Size of steps to take when varying
Decade	*Steps/Decade*	How many steps to take per decade
Octave	*Steps/Octave*	How many steps to take per octave
Logarithmic	*Total Steps*	Total number of steps to take
Times	*Multiplier*	Multiplier for taking steps **Note:** Typing `10` for *Multiplier* is tantamount to selecting *Decade* and specifying `1` step per decade.

6. (Optional) Click *Add Specification* to add another parameter specification.
7. Click *OK*.

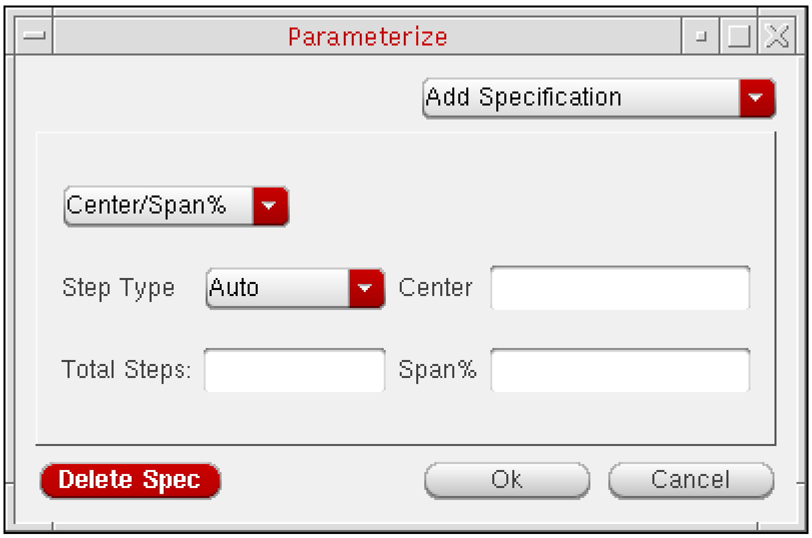
Note: You can change the specification type from *Center/Span* to *Center/Span%* or *From/To* by selecting a different item from the drop-down menu in the added section. See “Specifying Center and Span as a Percentage” on page 199 and “Specifying a Range of Values” on page 195 for more information.

Specifying Center and Span as a Percentage

To specify a center value and a span percentage, do the following:

1. From the *Add Specification* drop-down menu, choose *Center/Span%*.

   A section appears on the Parameterize form for defining the specification type you selected.



2. In the *Center* field, type the center value.
3. In the *Span* field, type a span value.

   The simulator can vary your parameter between $Center - \frac{(Span\% \times Center)}{100}$ and $Center + \frac{(Span\% \times Center)}{100}$ according to the *Step Type* you select (next).

4. From the *Step Type* drop-down menu, select how you want the simulator to vary the parameter value:

Linear	Linear steps from $Center - \frac{(Span\% \times Center)}{100}$ to $Center + \frac{(Span\% \times Center)}{100}$
Decade	Decade steps from $Center - \frac{(Span\% \times Center)}{100}$ to $Center + \frac{(Span\% \times Center)}{100}$

Octave	Octave steps from $Center - \frac{(Span\% \times Center)}{100}$ to $Center + \frac{(Span\% \times Center)}{100}$
Logarithmic	Logarithmic steps from $Center - \frac{(Span\% \times Center)}{100}$ to $Center + \frac{(Span\% \times Center)}{100}$
Times	Steps taken according to a specified multiplier from $Center - \frac{(Span\% \times Center)}{100}$ to $Center + \frac{(Span\% \times Center)}{100}$

5. Depending on the *Step Type* you selected, type the remaining value for your parameter specification in the field that appears:

Step Type	**Field that appears**	**Value**
Auto	*Total Steps*	Total number of steps to take
Linear	*Step Size*	Size of steps to take when varying
Decade	*Steps/Decade*	How many steps to take per decade
Octave	*Steps/Octave*	How many steps to take per octave
Logarithmic	*Total Steps*	Total number of steps to take
Times	*Multiplier*	Multiplier for taking steps **Note:** Typing `10` for *Multiplier* is tantamount to selecting *Decade* and specifying `1` step per decade.

6. (Optional) Click *Add Specification* to add another parameter specification.

7. Click *OK*.

Note: You can change the specification type from *Center/Span%* to *Center/Span* or *From/To* by selecting a different item from the drop-down menu in the added section. See "Specifying Center and Span" on page 197 and "Specifying a Range of Values" on page 195 for more information.

Deleting a Parameter Specification

To delete a parameter specification, do the following:

➤ On the Parameterize form, right-click in the row or section for the parameter specification you want to delete and choose *Delete Specification* from the pop-up menu. (*Delete Specification* is the only item on this pop-up menu.)

To delete the last parameter specification listed in the form, do the following:

1. On the Parameterize form, click *Delete Spec*.

   The last specification listed in the form is deleted.

2. Click *OK*.

Note: You can delete all but one specification in the form. If you want to delete the only remaining specification in the form, delete the values specified for the specification and click *OK*.

Creating Matched Device Parameters

You can create matched parameters for device matching such that one device's parameters track with another's. To specify two or more devices to match, do the following:

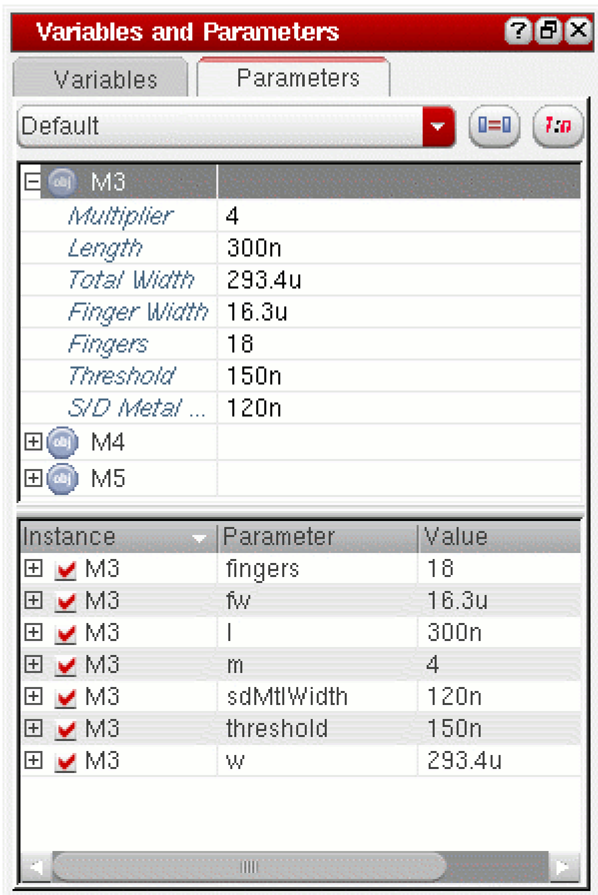
1. On your schematic, click to add the first device, then *Shift*+click to add each additional device.

   As you click, each device and its parameters appear on the upper half of the Parameters tab of the Variables and Parameters assistant pane.

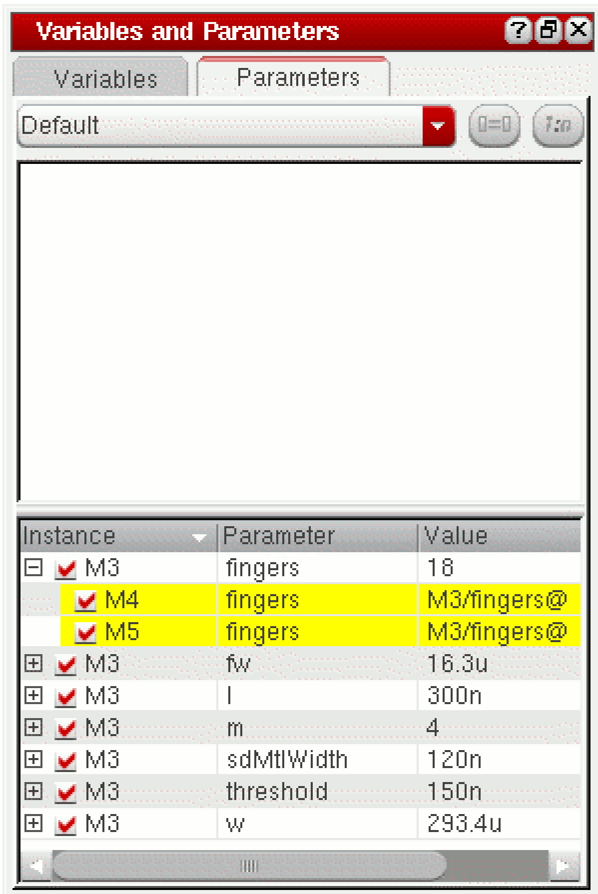
2. On the upper half of the Parameters tab of the Variables and Parameters assistant pane, select (highlight) the device you want to be the primary device against which all other devices are matched.

3. Click the *Match Parameters*  button.

The instance name of the primary device for which matched parameters exist appears on the lower half of the Parameters tab of the Variables and Parameters assistant pane.



4. (Optional) To view the devices whose parameters are matched to the primary device, click the plus sign to the left of an instance name on the lower half of the Parameters tab.



The value of a matched parameter appears in the *Value* column as follows to indicate the matching relationship:

`primaryInstName/deviceParam@`

For example, in the above figure, the value `M3/fingers@` for instance `M4` indicates that the `fingers` parameter of instance `M4` is matched to the `fingers` parameter of instance `M3` (the primary device).

Tip

You can also right-click on the lower half of the Parameters tab and choose *Expand All* to view the devices whose parameters are matched to all primary devices. Right-click and choose *Collapse All* to collapse the expanded view.

For information about using matched device parameters in the context of optimization, see the *Virtuoso Analog Design Environment GXL User Guide*.

Creating Ratio-Matched Device Parameters

You can create ratio-matched parameters for device tracking such that one device's parameters track with the ratio of another's. To specify two or more devices to ratio-match, do the following:

1. On your schematic, click to add the first device, then *Shift*+click to add each additional device.

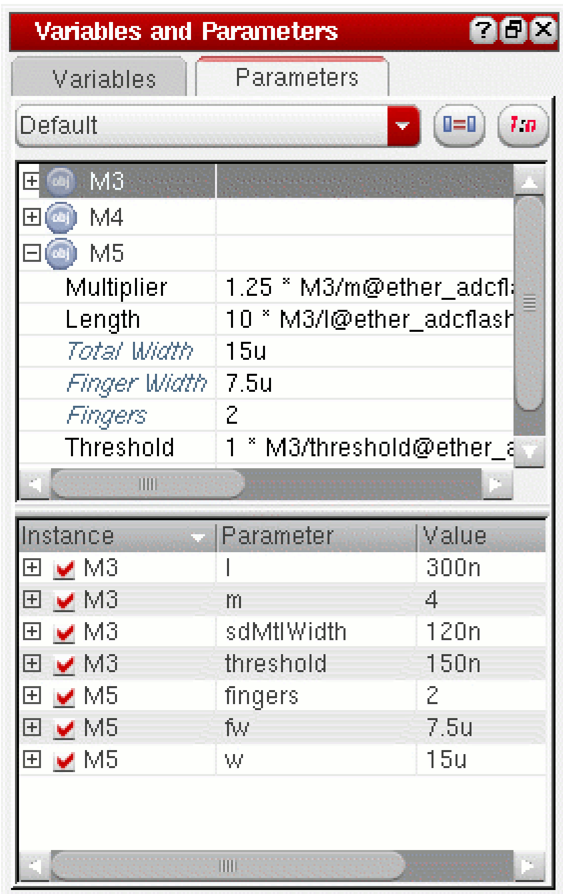
   As you click, each device and its parameters appears in the panel on the upper half of the Parameters tab of the Variables and Parameters assistant pane.

2. On the upper half of the Parameters tab of the Variables and Parameters assistant pane, select (highlight) the device you want to be the primary device against which all other devices are matched.

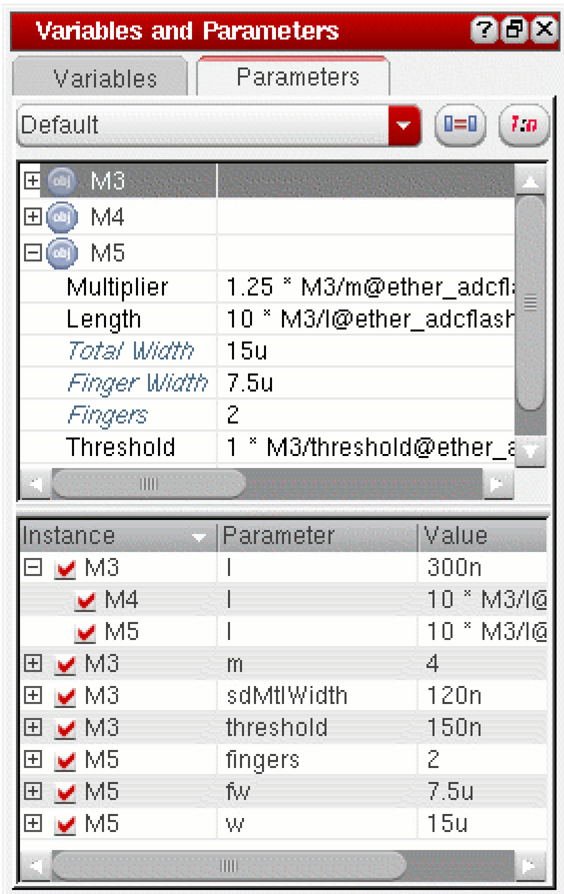
3. Click the *Ratio Matched Parameters*  button.

   The instance name of the primary device for which ratio-matched parameters exist appears on the lower half of the Parameters tab of the Variables and Parameters assistant pane.

Ratio-matched parameters appear in the panel on the lower half of the Parameters tab on the Variables and Parameters assistant pane.



4. (Optional) To view the devices whose parameters are ratio-matched to the primary device, click the plus sign to the left of an instance name on the lower half of the Parameters tab.



The value of a ratio-matched parameter appears in the *Value* column as follows to indicate the matching relationship:

*ratio*primaryInstName/deviceParam@*

For example, in the above figure, the value 10*`M3/l@` for instance `M4` indicates that the `l` parameter of instance `M4` is ratio-matched to the `l` parameter of instance `M3` (the primary device).

Tip

You can also right-click on the lower half of the Parameters tab and choose *Expand All* to view the devices whose parameters are matched to all primary devices. Right-click and choose *Collapse All* to collapse the expanded view.

For information about using ratio-matched device parameters in the context of optimization, see the *Virtuoso Analog Design Environment GXL User Guide*.

Creating a Combinatorial Expression

A combinatorial expression is one created using more than one output from one or more tests. For example, you might want to create an expression such as

`Test1_Output1/Test2_Output2`

You can create combinatorial expressions as values for a global variable.

To create a combinatorial expression such as the one above, do the following:

1. On the Variables tab of the Variables and Parameters pane, right-click a test name and choose *Add Variable*.

   The Editing Design Variables form appears.

2. In the *Name* field, type a name for the variable (such as `Out1DivOut2`).
3. On the Outputs Setup tab, drag an output from any test and drop it in the *Value (Expr)* field on the Editing Design Variables form.
4. Type `/` at the end of the string in the *Value (Expr)* field.
5. On the Outputs Setup tab, drag another output from another test and drop it at the end of the expression in the *Value (Expr)* field on the Editing Design Variables form.
6. Click *OK*.

   The new variable appears in the *Global Variables* tree on the Data View pane and the Variables tab of the Variables and Parameters pane.

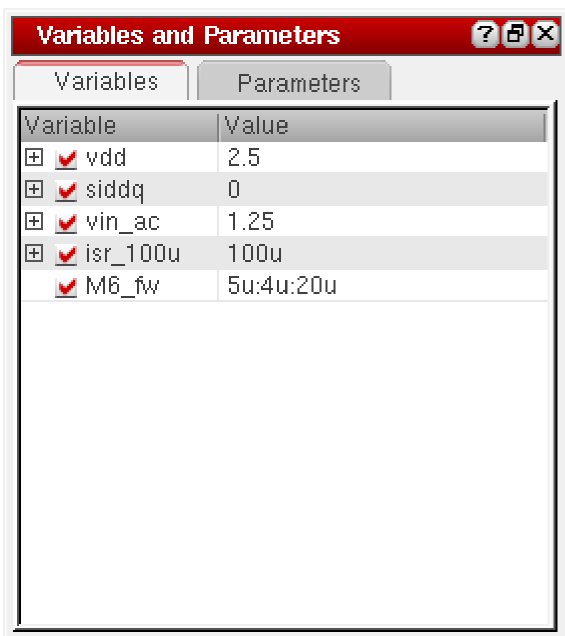
   Note: The expression uses the `calcVal` function.

Toggling the View on the Variables tab of the Variables and Parameters Assistant Pane

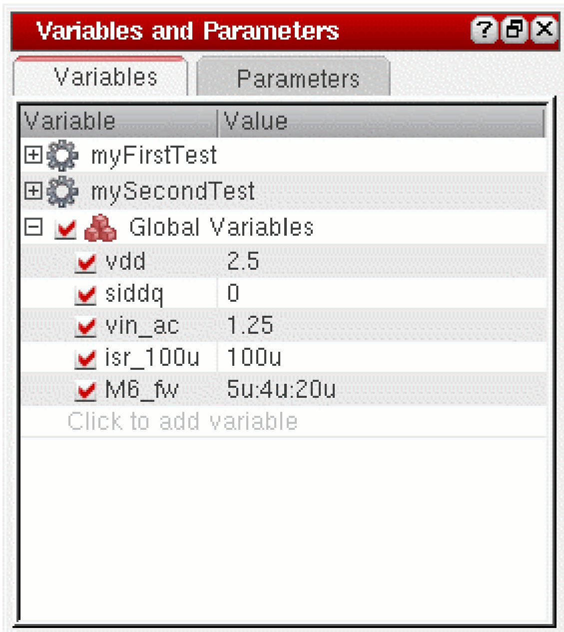
To toggle the view of variables on the Variables tab of the Variables and Parameters assistant pane, do the following:

➤ Right-click any item and choose *Toggle View*.

If you were viewing the variable categories, the view changes to a tree of variables.



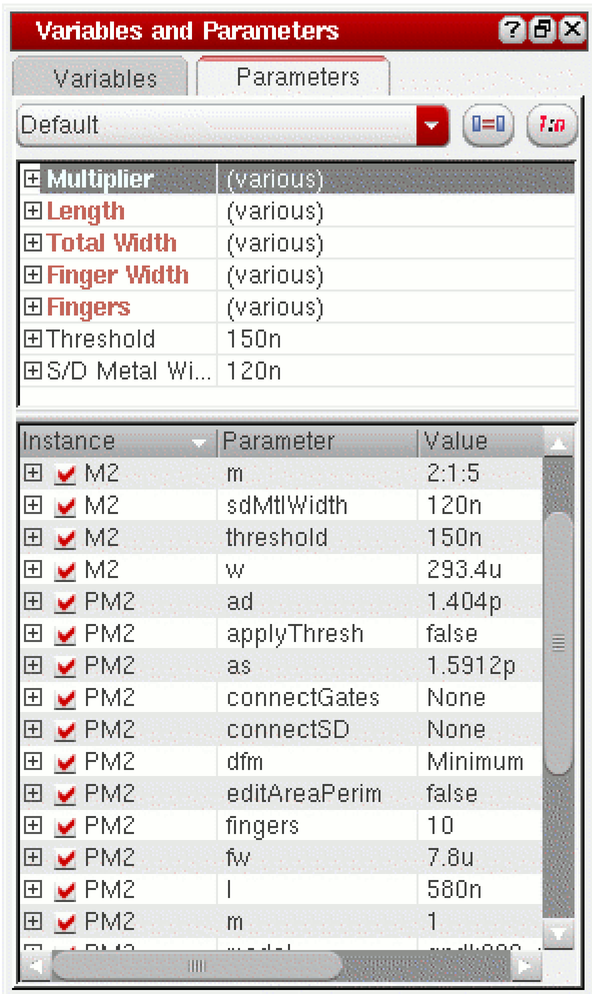
If you were viewing a tree of variables, the view changes to the variable categories view.



Sorting Parameters by Properties and Objects

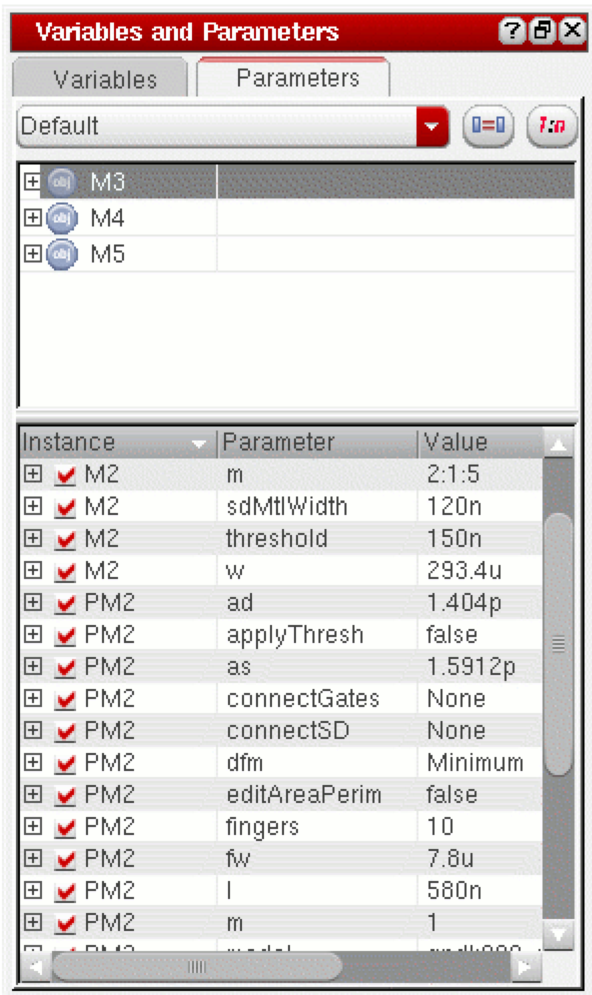
To sort the parameters on the upper part of the Parameters tab of the Variables and Parameters assistant pane by properties, do the following:

➡ Right-click on the upper part of the Parameters tab and choose *View by Property*. The view changes to a tree of properties.



To sort the parameters on the upper part of the Parameters tab of the Variables and Parameters assistant pane by objects, do the following:

➡ Right-click on the upper part of the Parameters tab and choose *View by Object*. The view changes to a tree of objects.



Disabling Callbacks on Swept Device Parameters

By default, the program executes callbacks on device parameters that you sweep.

To disable callbacks on swept device parameters, do the following:

➤ In your `.cdsinit` file, set the `axlExecuteCallbacks` variable to `nil`:

```
axlExecuteCallbacks=nil
```

For more information about callbacks in parameter expressions, see also

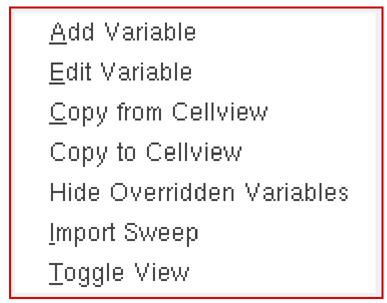
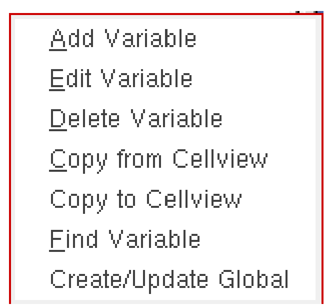
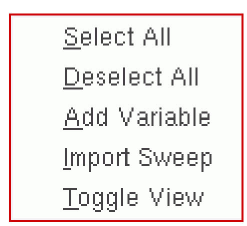
- “Scope of Parameters” in the *Virtuoso Analog Design Environment L User Guide*

- "Triggering callbacks" in the "Advice and Warnings" appendix of the *Component Description Format User Guide*

Variables and Parameters Assistant Right-Click Menus

Right-click menus are available when you have instance parameters that appear on the upper half of the Parameters tab on the Variables and Parameters assistant pane and for design variables and *Global Variables* that appear on the Variables tab of the pane.

For design variables and *Global Variables* that appear on the Variables tab of the Variables and Parameters assistant pane, the right-click menus you see depend on what the view is and what you right-click as follows:

View	Right-Click	Menu
Variables categories view	Test name (tree heading)	
	Design variable (leaf)	
	Global Variables (tree heading)	

View	**Right-Click**	**Menu**
	Global variable (leaf)	
Variables tree view	*Global Variables* check box (leaf)	
	Test name (leaf)	

For instance parameters that appear on the upper half of the Parameters tab on the Variables and Parameters assistant pane (when you select one or more devices on the schematic), the right-click menus you see depend on what the view is and what you right-click as follows:

View	Right-Click	Menu Choice(s)
Parameters by instance	Instance name (tree heading)	*Select All* *View by Property* *View by Object*
	Instance parameter (leaf)	*Create Parameter* *Create Parameter Range* *Set to Design Value* *Select All* *View by Property* *View by Object*
Parameters with instances as leaves	Instance parameter (tree heading)	*Select All* *View by Property* *View by Object*
	Instance name (leaf)	*Create Parameter* *Select All* *View by Property* *View by Object*

5

Working with Constraints

ADE XL supports the matched parameter and correlation constraints you specify in Virtuoso Constraint Manager.

- ADE XL creates parameters for the matched parameter constraints specified in Constraint Manager. These parameters are displayed on the `Data View` and the Parameters tab on the Variables and Parameters assistant panes. For more information about working with parameters created for matched parameter constraints, see Working with Parameters Created for Matched Parameter Constraints on page 218.
- Correlation constraints are used for Monte Carlo analysis. Correlation constraints are not displayed on the Data View or the Variables tab of the Variables and Parameters assistant panes—they are written to the netlist when you run a Monte Carlo analysis. For more information about Monte Carlo analysis, see Chapter 7, "Performing Monte Carlo Analysis."

For more information, see the following topics:

Adding, Modifying, and Deleting Constraints

To add, modify or delete matched parameter and correlation constraints in Constraint Manager, do the following:

1. On the Data View assistant pane, right-click a test and choose *Open Design in Tab*.

   The schematic design associated with the test appears on a new tab in your session window.

2. Choose *Window – Assistants – Constraint Manager*.

   The Constraint Manager pane appears.

3. Add, modify or delete matched parameter and correlation constraints. For more information about using Constraint Manager, see the *Virtuoso Unified Custom Constraints User Guide*.

4. Click the *Save* button in Constraint Manager.

5. Click the *Check and Save* button in the Schematic editor.

Working with Parameters Created for Matched Parameter Constraints

When you launch ADE XL, parameters are automatically created for the matched parameter constraints specified in Constraint Manager. These parameters are displayed in the *Parameters* tree on the Data View pane and the Variables tab of the Variables and Parameters assistant pane.

Figure 5-1 Matched Parameter Constraint Defined in Constraint Manager

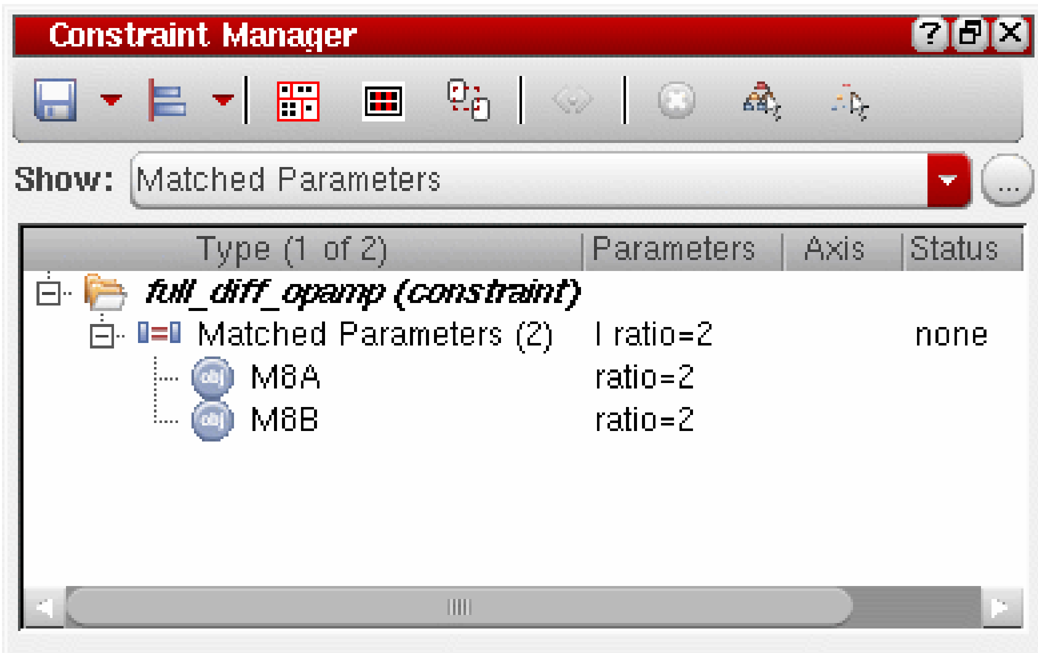
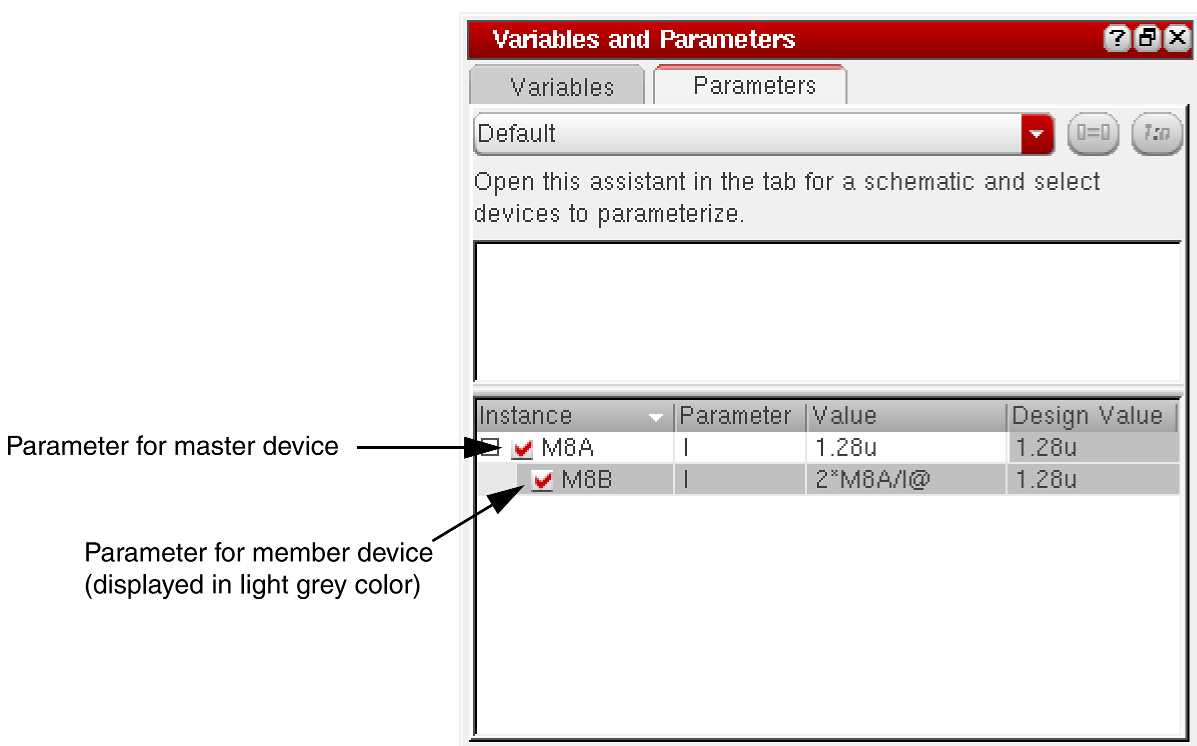


Figure 5-2 Corresponding Parameters in ADE XL for Matched Parameter Constraints

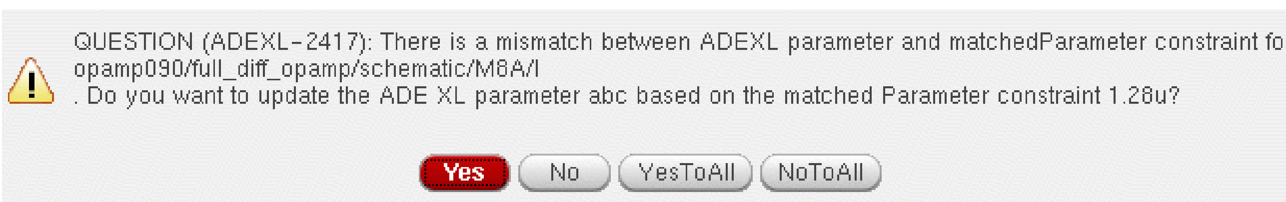


Note the following:

- In the Parameters tab on the Variables and Parameters assistant pane, you cannot modify or delete the matched parameter constraints that are created by Constraint Manager.
- If you make changes to matched parameter constraints in Constraint Manager while ADE XL is running, the changes are automatically reflected in ADE XL. For example, if you modify a matched parameter constraint in Constraint Manager while ADE XL is running, the corresponding parameters in ADE XL are also automatically modified.

Resolving Mismatch Between Matched Parameter Constraints and their Corresponding Parameters

When you launch ADE XL, or restore a history item in ADE XL, ADE XL checks if there is any mismatch between the matched parameter constraints in Constraint Manager and their corresponding parameters in ADE XL. The following message is displayed for each mismatch:



Do one of the following:

- Click *Yes* to update the parameter based on the corresponding matched parameter constraint.
- Click *No* to ignore the mismatch and mark the corresponding matched parameter constraint as *impossible* in Constraint Manager.
- Click *Yes to All* to update all mismatched parameters based on their corresponding matched parameter constraints.
- Click *No to All* to ignore all the instances where there is a mismatch and mark the corresponding matched parameter constraints as *impossible* in Constraint Manager.

6

Simulating Corners

A corner is a combination of variables or process models that define a scenario in which you want to measure the performance of your design. In ADE XL, you can create corners using the Corners Setup form where you can vary the values for temperature, parameters, design variables, and model files from any test. You can then run the corners for one or more tests and measure how the tests perform in these varying conditions. If the corner settings are already available in pre-defined files, you can also load and use them to create corners.

This chapter describes the Corners Setup form and the procedures to create and use corners in ADE XL.

See the following topics for more details:

- Opening the Corners Setup Form on page 222
- Adding Corners on page 226
- Modifying Corner Values on page 239
- Working with Corners on page 241
- Working with Corner Groups on page 257
- Simulating Corner on page 262

Video

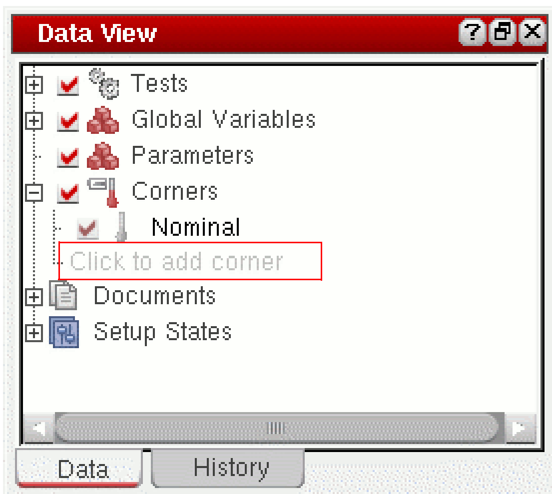
You can view video demonstration for this feature at Using the New Corners Setup Form in ADE XL.

Opening the Corners Setup Form

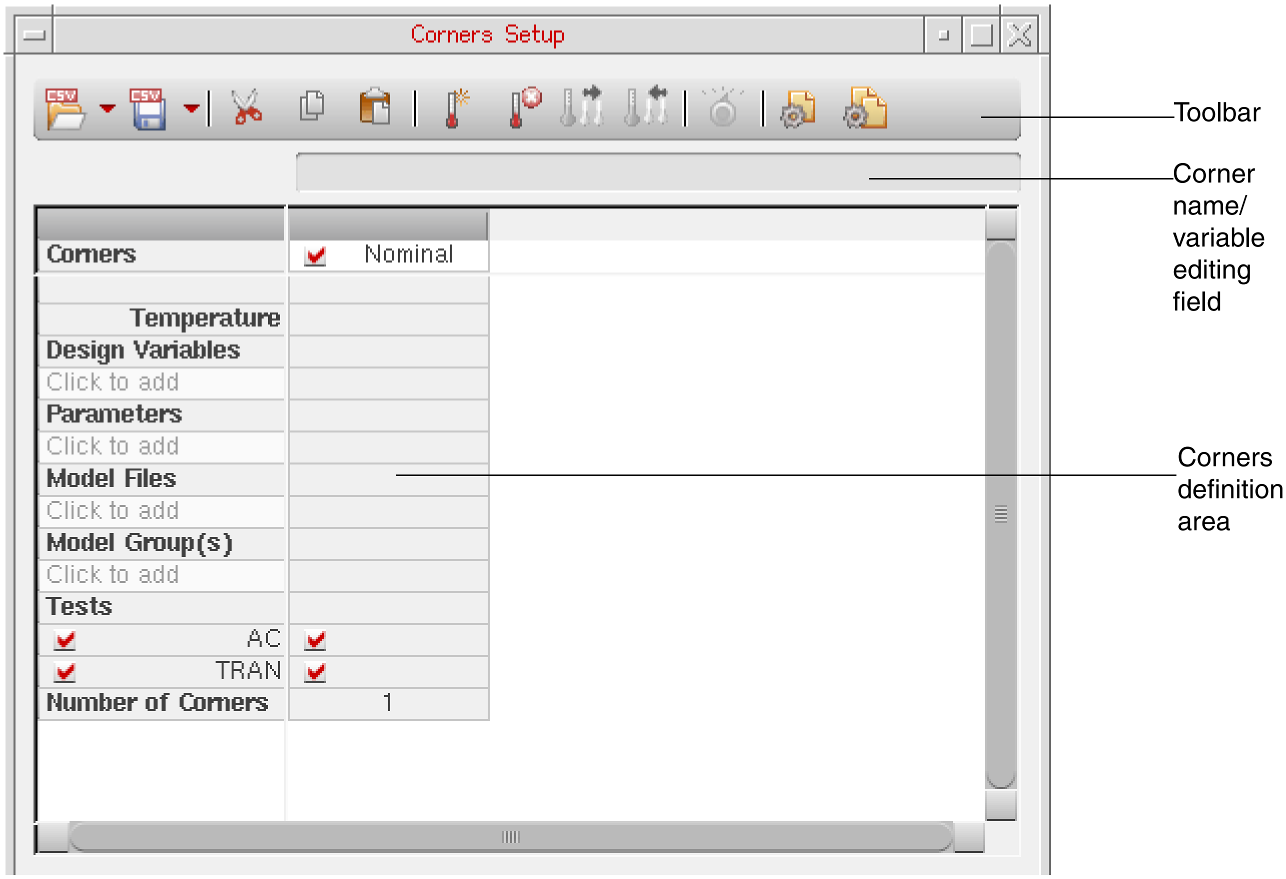
You can add or remove corners or modify corner settings by using the Corners Setup form.

To open the Corners Setup form, do one of the following:

- Choose *Create – Corners*.
- On the Create toolbar, click .
- In the Data View pane, expand *Corners* and click where it says *Click to add corner*.



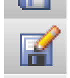
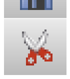
The Corners Setup form is displayed.

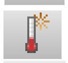
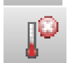
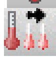
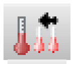
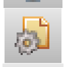


Note: Each corner is added as a new column on this form. By default, the table contains a default corner, named `Nominal`. This is the nominal corner and runs without varying the corners variables.

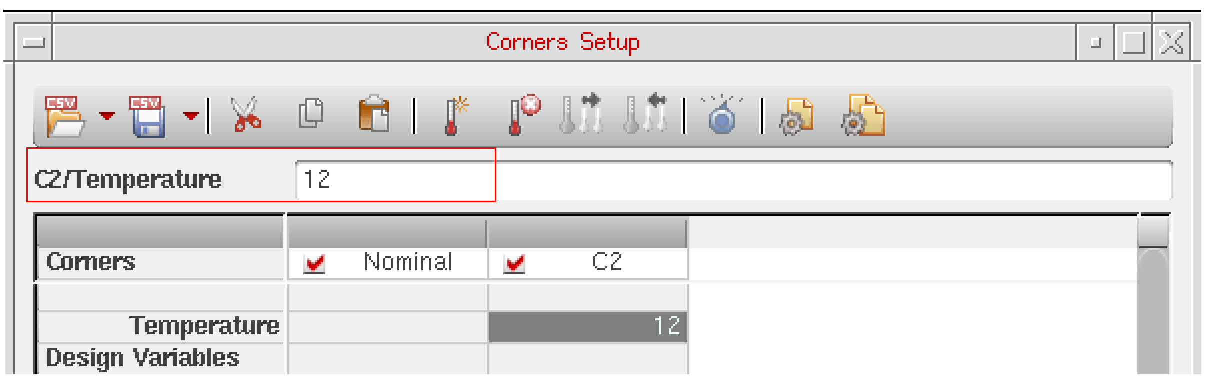
The Corner Setup form contains:

- A toolbar that provides the following commands:

Toolbar button	Description
	Loads corners from a setup database file
	Imports corners from a Process Customization Files (PCF) or Design Customization Files (DCF).
	Saves corners to the active setup
	Imports corners from an external file
	Moves the selected text to the clipboard

	Copies the selected text to the clipboard
	Pastes the copied text from the clipboard
	Creates a new corner
	Deletes the currently active corner
	Expands a corner group
	Creates a corner group
	Sets a sweep value for the variable being edited
	Adds or edits a model file
	Adds or edits a model group

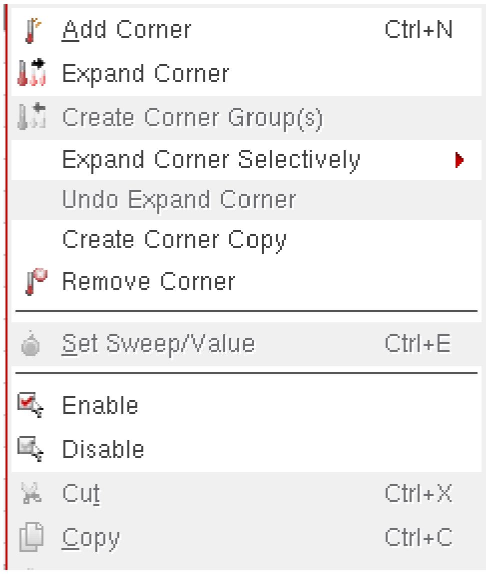
- The variable editing text field: In this field, you can edit values for corner variables. The label to the left of this field changes dynamically to show the name of the corner and the variable that you are currently editing. For example, if you click in a cell that defines temperature for corner `C2`, the label for this field is shown as *C2/Temperature*, as shown in the following figure.



- Corner definition area: Displays the corner definitions in a tabular format. The first column in the table lists the variables organized into various groups. Each subsequent column contains the definition of a corner where you can edit the values corresponding to these variables. When you add a corner, a new column is created for it. The top row in the table displays the corner names.

  You can add variables to a corner definition by clicking where it says *Click to add*. You will learn more about this in the following sections.

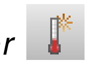
- Pop-up menu: The most frequently-used commands for corners are available in the pop-up menu of the Corners Setup form. When you right-click in a corner column, the following pop-up menu is displayed:



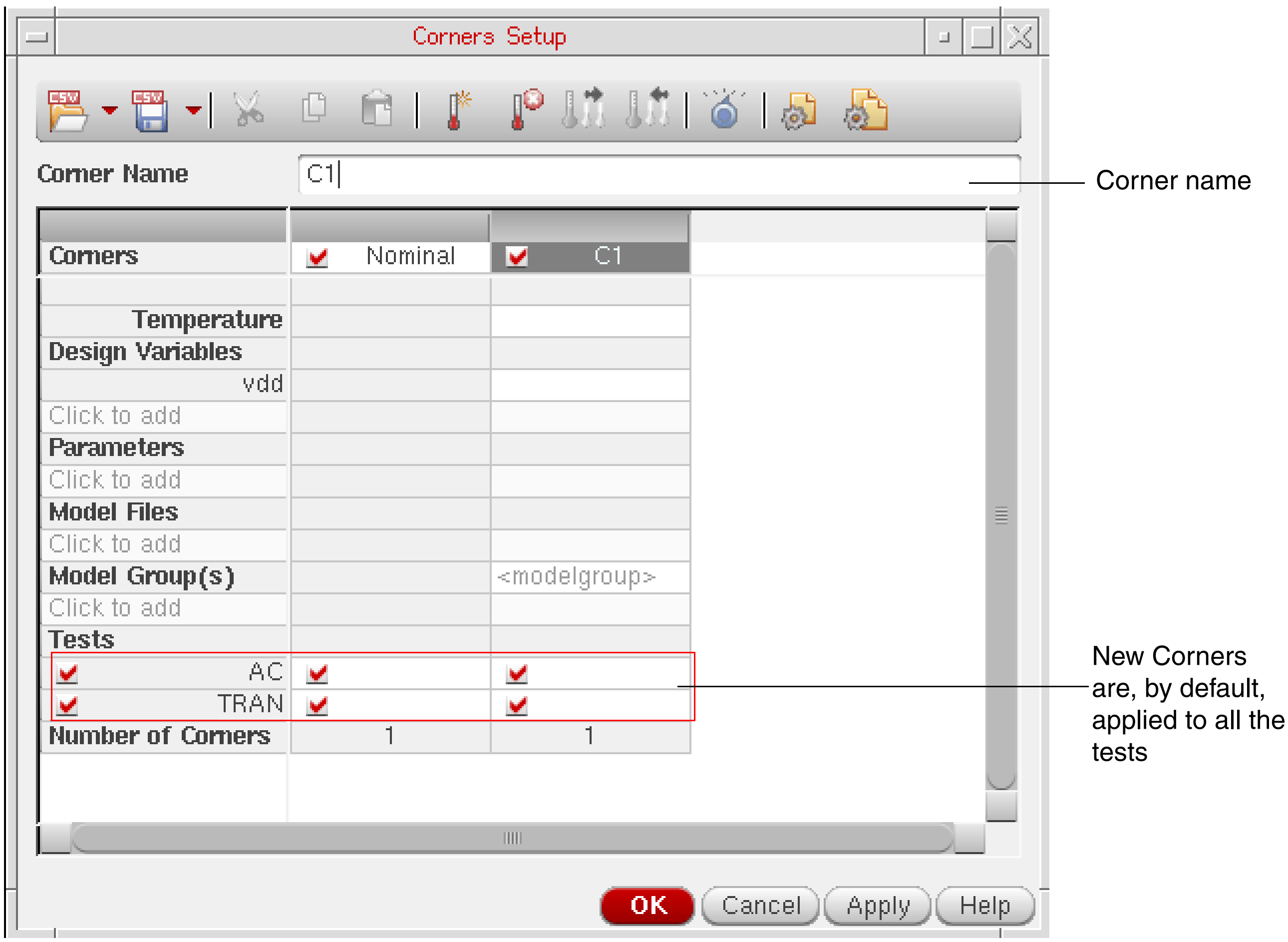
For more details about how to add a corner or specify variables, refer to the following sections.

Adding Corners

You can add a corner to define the conditions under which you want to check a design of your tests.

To add a new corner, in the Corners Setup form, click *Add new corner*  on the toolbar.

A new column is added in the Corners Setup form, as shown in the following figure.



By default, the corners are named sequentially, prefixed with `C`, such as `C1`, `C2`, and so on. You can change the names, if required.

After adding a new corner, specify values for the variables that define the corner conditions.

Note: In a corner column, all the text fields, for example, Temperature or a design variable, are by default edited in the append mode. When you click on a field and type a value, it is appended to the already existing value, if any. To replace the existing value, select and delete it and enter a new value.

You can specify one or more of the following settings for a particular corner:

- Specify temperature
- Add design variables
- Add parameters
- Add model files
- Specify sections of model files
- Add model groups

Important

In addition to these corners, you can also create statistical corners based on the results for a corner in a Monte Carlo sample. To know more about what statistical corners are and how to create these corners from the Monte Carlo results, refer to Managing Monte Carlo Results in the Yield View.

In addition, you may want to refer to the following sections to know how to perform the various tasks related to corners:

- Modifying Corner Values on page 239
- Renaming a Corner on page 241
- Disabling and Enabling Corners on page 242
- Viewing Corner Settings on page 246
- Copying Corners on page 246
- Exporting Corners on page 247
- Importing Corners on page 249
- Importing Corners from Customization Files on page 253
- Setting Up a Default Set of Corners on page 255
- Creating a Corner Group on page 257

Specifying Temperature

To specify the temperature for a corner, do the following in the column for that corner:

➡ Click the cell next to the *Temperature* label and type a value.

Note: You cannot specify a value if a parameter or variable named temperature is specified in your ADE XL test setup.

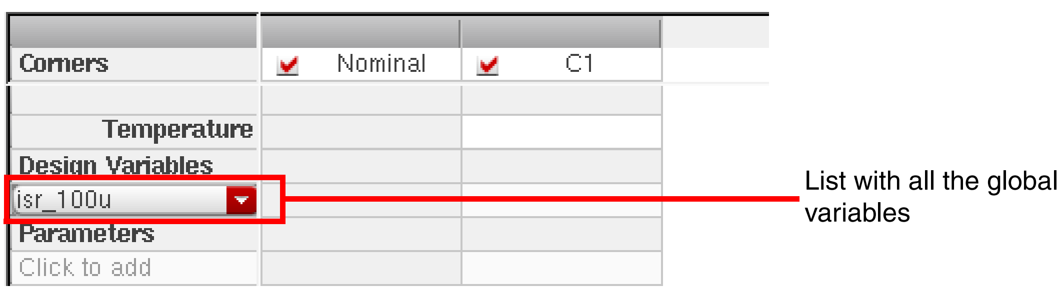
Return to adding corners procedure.

Specifying Values for Design Variables and Parameters

To specify the value for a design variable for a corner, do the following in the column for that corner:

1. Under *Design Variables*, click *Click to add*.

   A drop-down list with all the global variables is displayed.



2. From this list, select the name of the variable that you want to add to the corner.
3. To specify the value for this variable, click in the cell corresponding to this variable in the column that contains the corner definition.

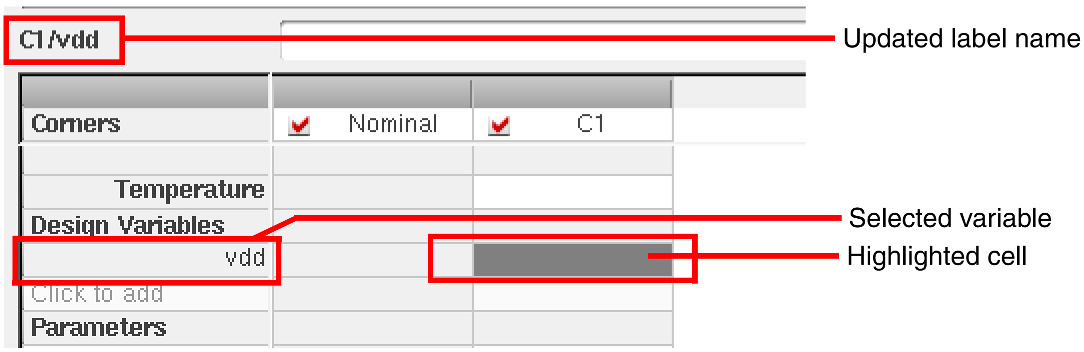
   Observe the following changes:

   ❑ The cell that you clicked is highlighted.

   ❑ The label for the corner name/variable editing field changes to <corner-name>/<variable-name>.

   ❑ A new blank row, the label for which reads *Click to add*, is added below the variable you just created.

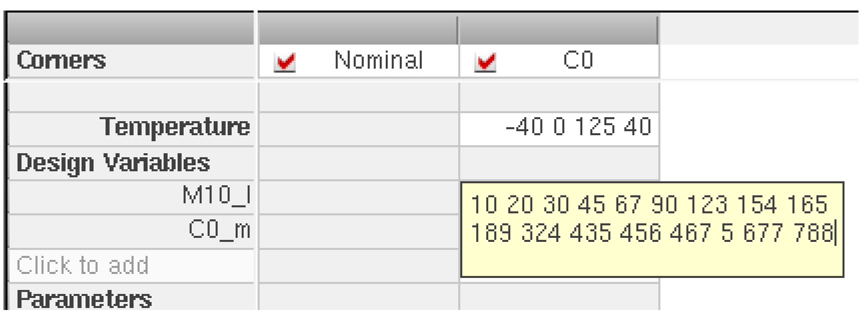
For example, when you select the `vdd` variable from the list, you will see the changes highlighted in the following figure.



4. Specify value for the variable in the corner name/variable editing field.

You can either specify a single value, a space- or comma-separated set of sweep points, or a range of values in the *`startValue:increment:stopValue`* format.

If you specify a long list of sweep points for a variable, the tool automatically expands the text field up to the right margin of the form so that all the values can be viewed while editing, as shown in the following figure.



Note: When you specify sweep points for a variable, the number of corners are calculated based on the number of sweep points for all the variables, parameters, or model files specified for that corner. For example, if for a corner, two variables have two sweep points each, the total number of corners is four. To see the setup for each corner, click  on the toolbar.

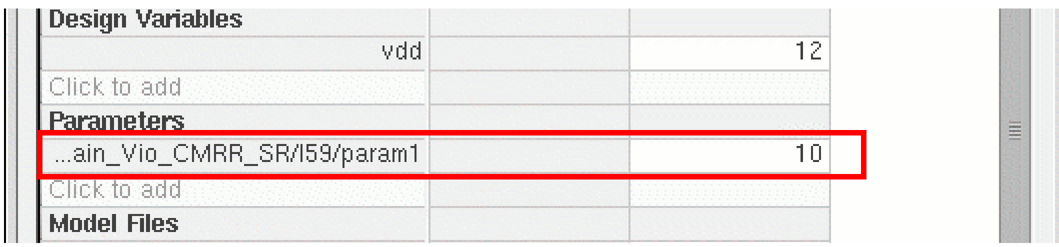
Important

By default, before running a simulation, ADE XL does not match the design variables in the Corners Setup form with the list of global variables in the active setup. If the setup for corners uses any design variable that is not present in the active ADE XL setup, a redundant simulation is run for that variable. To ensure that any design variable that does not exist in the ADE XL setup is reported as an error before running a simulation, set the showErrorForNonExistingVariables environment variable to `t`.

Return to adding corners procedure.

Specifying Values for Parameters

For a corner, you can specify parameters in the same way as you specify variables. When you click *Click to add* under *Parameters*, a drop-down list that contains the names of all the CPF parameters created for the cells or instances used in the tests is displayed. Select the required parameter from this list and specify a value for the parameter.



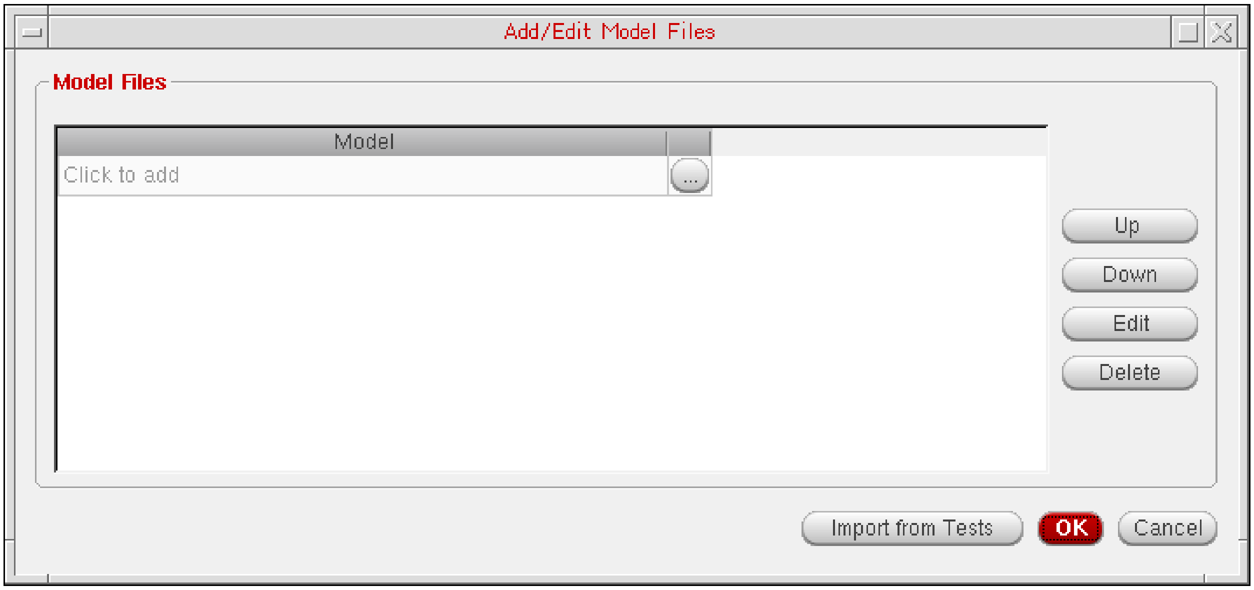
Return to adding corners procedure.

Adding Model Files to a Corner

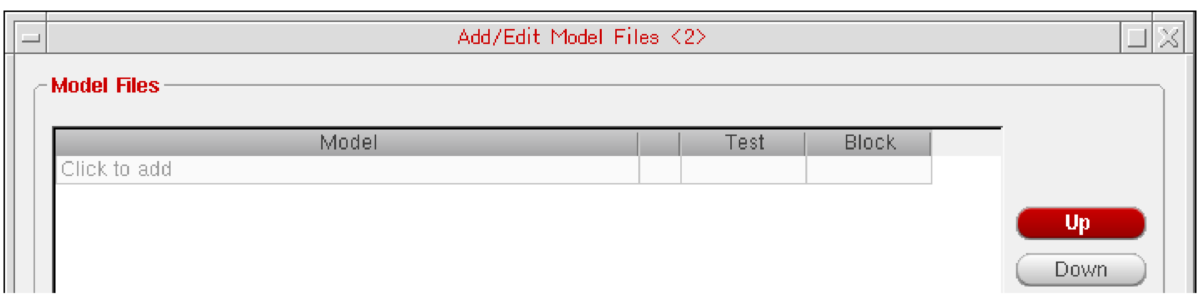
For each model file you want to add as a corner, do the following in the Corners Setup form:

1. In the Corners Setup form, do one of the following:
   - ❑ On the toolbar, click .
   - ❑ Under *Model Files*, click where it says *Click to add*.

The Add/Edit Model Files form is displayed.

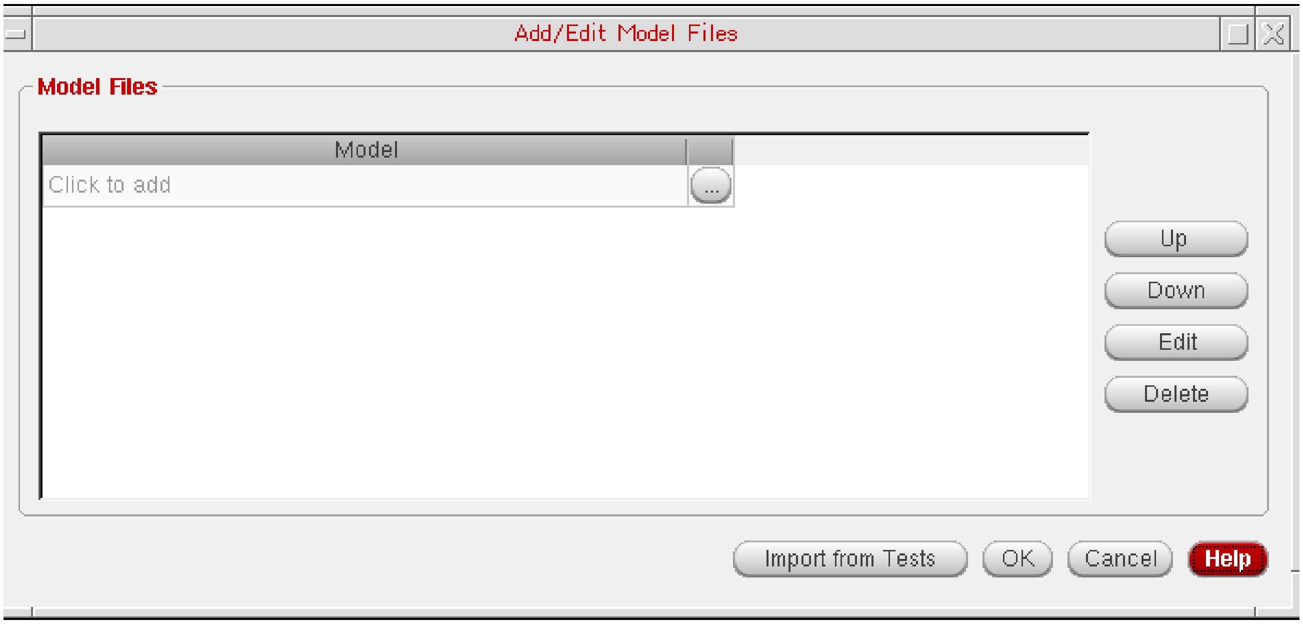


Note: If you have enabled MTS, *Test* and *Block* columns also appear in the form. For more details about MTS, refer to *Virtuoso Analog Design Environment GXL User Guide*.



2. In the *Model Files* group box, do one of the following to add model files:
   - ❑ Click *Click to add* and type the path to and name of a model file.
   - ❑ Click the browse button to select a model file.
   - ❑ Click *Import from Tests* to import model files from the tests in your setup.

All the model files specified for tests in your setup are displayed in the form. For more information about specifying model files for tests, see Specifying Model Libraries.



In addition to selecting the model files to be used for a corner, you can also do the following in the Add/Edit Model Files form:

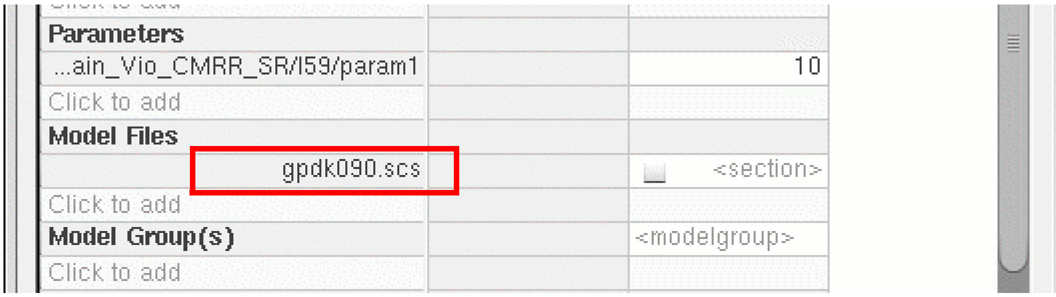
- Modify the search order that the simulator uses to find simulation models in the model files: By default, the simulator searches for models in the model files in the order in which they are listed in the Corners Setup form. The simulator uses the first simulation model it finds with the matching name. To modify the search order, select the model file and click *Up* or *Down*. The selected model file moves up or down in the list.

- Edit a model file: Select the model file you want to edit and click *Edit*. The model file is displayed in a text editor. Make the changes you want to make. Save your changes and close the text editor.

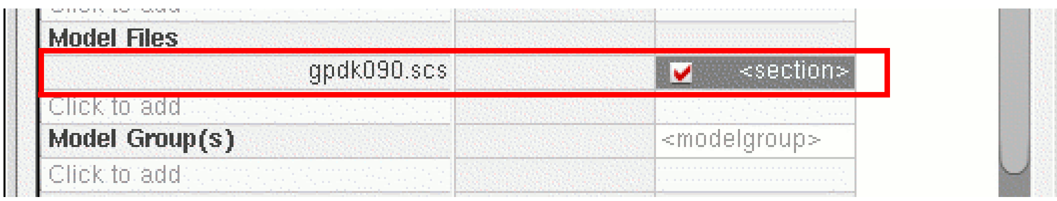
- Remove a model file from the list: Select the model file and click *Delete*.

3. Click *OK*.

The model file name is displayed in the Corners Setup form.



4. In the column for a corner, select the check box in the cell corresponding to the model file to add it to the corner.



If required, you can specify a selected set of sections of the model file. For more details on how to specify sections for a model file, refer to Specifying Sections for Model Files.

Important

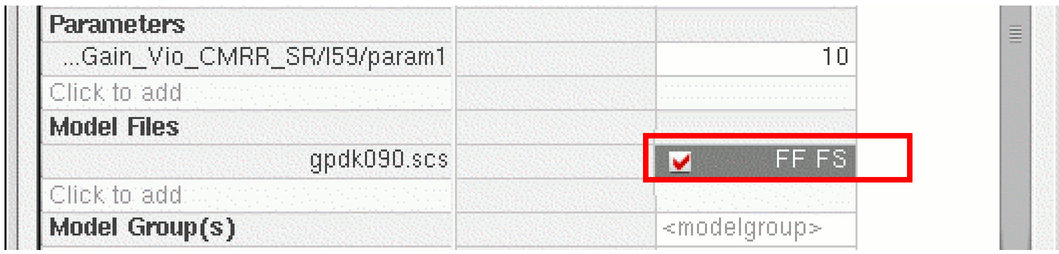
If you have imported model files from a test, the tool does not import the corresponding section names to be used. In this case, while running the corner simulation for a test, the tool considers those sections of the model file that are specified for the test. However, if required, you can specify a different set of section names.

Return to adding corners procedure.

Specifying Sections for Model Files

When you specify a model file for a corner, in the column for that corner, the cell corresponding to a model file shows a default text *<section>*. This implies that the corner uses all the sections defined in the specified model file. If required, you can specify a selected set of sections of the model file that you want to use for the corner.

To specify sections, double-click in the model file cell for the corner. A drop-down list that contains the names of all the sections defined in the file is displayed. Select a section that you want to include in the corner. You can also select more than one section by selecting the required sections one by one. Each section that you select is displayed in a space-separated list, as shown in the following figure.



Note the following:

- If you do not have the read permission for the model file, the section names will not be displayed in the drop-down list.
- If the same model file is specified in the Model Library Setup form for more than one test, but different sections are specified for each test, only the sections that are common to all the tests that are enabled in the Data View pane will be displayed in the drop-down list. For more information about specifying model files for tests, see Specifying Model Libraries on page 96.
- If you know the section names, you can directly edit the field and enter the names. However, if the specified section name does not exist in the corresponding model file, simulation for the corner would fail. To avoid this, you can set the LimitModelSections environment variable to `InModelFile` to specify that if the section name does not exist in the model file, the tool should display an error message in the Corners Setup form.

Return to adding corners procedure.

See also

- Varying the Model File and Section during Simulation on page 359.

Adding a Model Group to a Corner

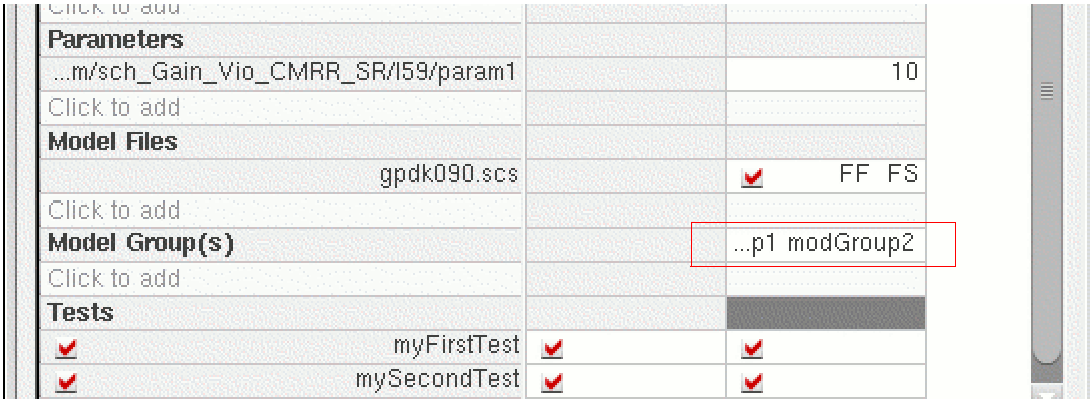
To add one or more model groups to a corner, do the following in the Corners Setup form:

1. In the column for a corner, double-click where it says *<modelgroup>*.

   A drop-down list with all the available model groups is displayed in this cell.

2. Select one by one all the model groups that you want to add to the corner.

Each model group that you select is displayed in a space-separated list in the column for the corner.



Note: For every model group that you add, a separate corner simulation will run.

See also

- Creating a Model Group on page 235
- Saving and Loading Model Groups on page 237

Return to adding corners procedure.

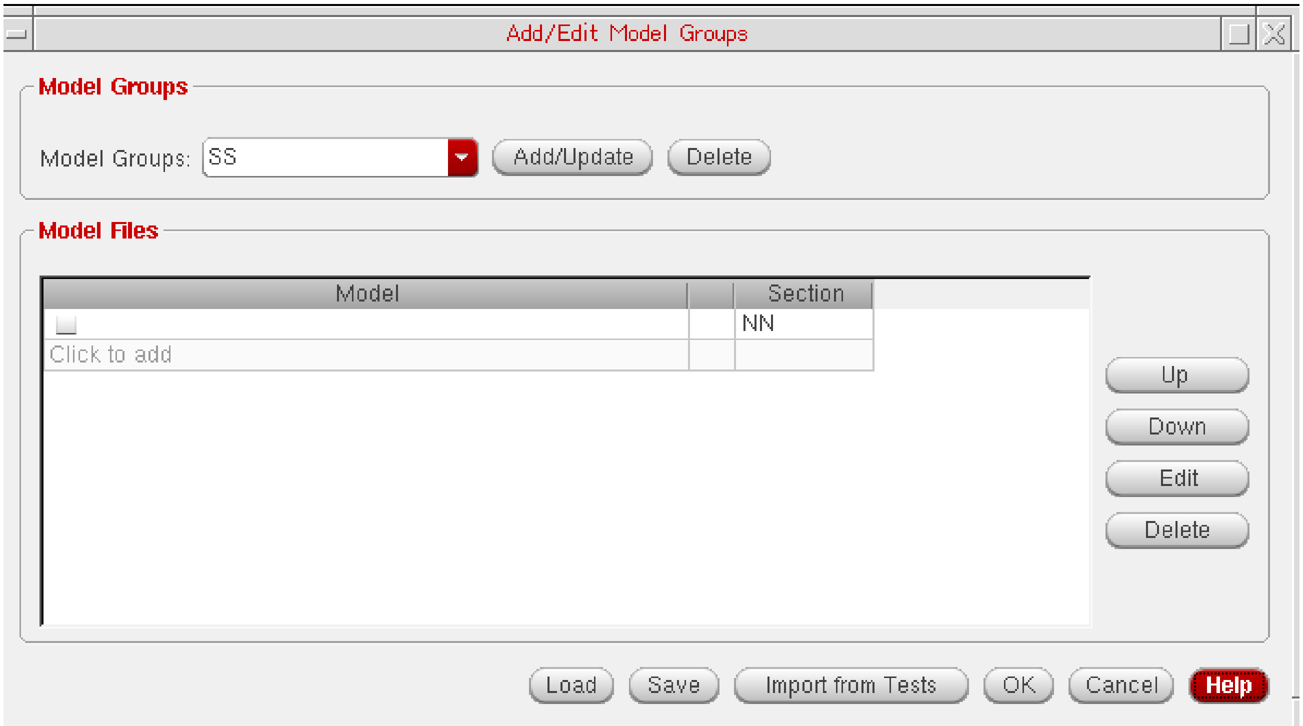
Creating a Model Group

A model group contains model files that you can use when simulating a corner. For each model file in the group, you can also specify selected sections to be used.

To create a model group, do the following in the Corners Setup form:

1. Do one of the following:
   - ❑ Click the *Add/Edit Model Group(s)*  button.
   - ❑ Under *Model Group(s)*, click where it says *Click to add*.

The Add/Edit Model Groups form is displayed.



2. In the *Model Groups* field, type a name for the model group.

3. Click the *Add/Update* button.

4. Add one or more model files to the model group in one of the following ways:

- In the *Model* column of the *Model Files* table, click *Click to add* and type a valid path to a model file.
- In the *Model Files* table, click the browse button to open the Choose Model File form. Browse to select a model file and click *Open*.
- Click *Import from Tests*. Model files are imported from all the tests and are displayed in the *Model Files* table. By default, the files appear enabled. You can disable a model file in a model group by deselecting the check box to the left of the model file.

5. (Optional) By default, the simulator searches for models in all the sections of a model file. If required, you can specify one or more sections from each model file by double-clicking in the *Section* column and selecting the required sections from the drop-down list that is displayed.

6. Click *OK*.

In addition to creating a model group, you can also do the following using the Add/Edit Model Groups form:

- Remove a model file from the model group: Select the model file in the *Model Files* table and click *Delete*.
- Edit a model file: Select the model file in the *Model Files* table and click *Edit*.
- Modify the search order for the model files: Select the model file in the *Model Files* table and click *Up* or *Down*.
- Create a new model group from an existing model group: You can do this by performing the following steps in the Add/Edit Model Groups form:

  a. In the *Model Groups* field, select an existing model group from the drop-down list.

  Model files in that model group appear in the *Model Files* table.

  b. If required, make the changes you want for the new model group. You can add or remove model files or their sections. You can also edit a file or change the search order.

  c. In the *Model Groups* field, type a new name for the model group.

  d. Click *Add/Update*.

  A new model group is created with the name that you specified and this name is also added to model groups drop-down list in the Corners Setup form.

Saving and Loading Model Groups

You can save the model group information in the Add/Edit Model Groups form to a file and use the file later to load the model group information in the same or a different ADE XL view.

Saving a Model Group

To save model group information, perform the following steps:

1. Click *Save*. The Save Model Groups form is displayed.
2. Browse to locate the folder in which to save the file and type a name for the file in the *File name* field.
3. Click *Save*.

The model group information is saved in the XML format. The file has the `.sdb` extension.

Loading a Model Group

To load model group information from a previously saved file, perform the following steps:

1. Click *Load*. The Load Model Groups form is displayed.
2. Browse to locate the required model group information file.
3. Click *Open*.

The model group information is displayed in the Add/Edit Model Groups form.

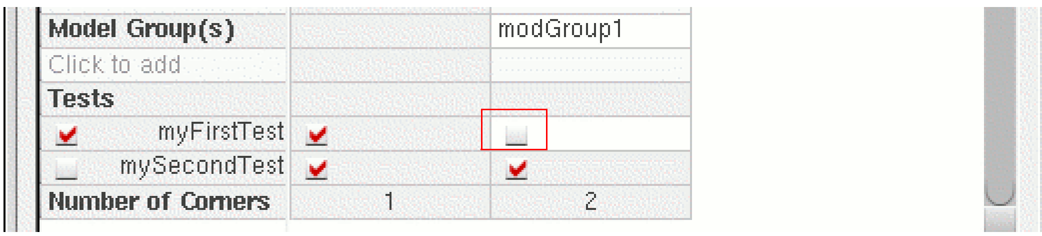
Return to adding corners procedure.

Modifying Corner Values

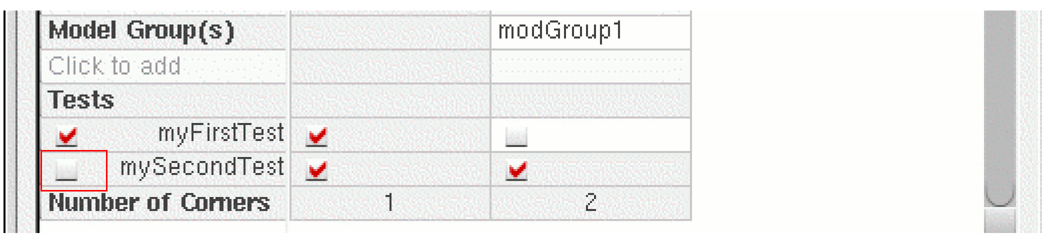
After you add corners, you can modify values for individual corners in the Corners Setup form. To edit the values of design variables or parameters, select the cell to display its value in the variable editing text field. You can edit the value in this field. Alternatively, double-click in the cell containing that value and edit the value there.

In addition, to editing values, you can also do any the following to modify a corner:

- You can enable/disable a model for a corner: For this, select/deselect the check box next to the model name in the column for the corner.
- You can enable/disable a corner for a test: For this, select/deselect the check box next to the test name in the column for the corner.



- You can enable/disable all corners for a test: For this, select/deselect the check box to the left of the test name.



- You can remove a design variable, parameter, or model file from the Corners Setup form. For this, select one or more items that you want to remove, right-click and choose *Remove*.

Important

If the Corners Setup form is already open and you change the corner or test details in the setup database, the changes are automatically updated in this form if the autoCornerUpdate environment variable is set to `t`.

Also see:

- Disabling and Enabling One or More Corners for All Tests on page 243

Working with Corners

This section describes the various tasks that you can perform while working with the corners:

Renaming a Corner

By default, the program names corners as `C1`, `C2`, and so on. To change the default name of a corner, do any of the following:

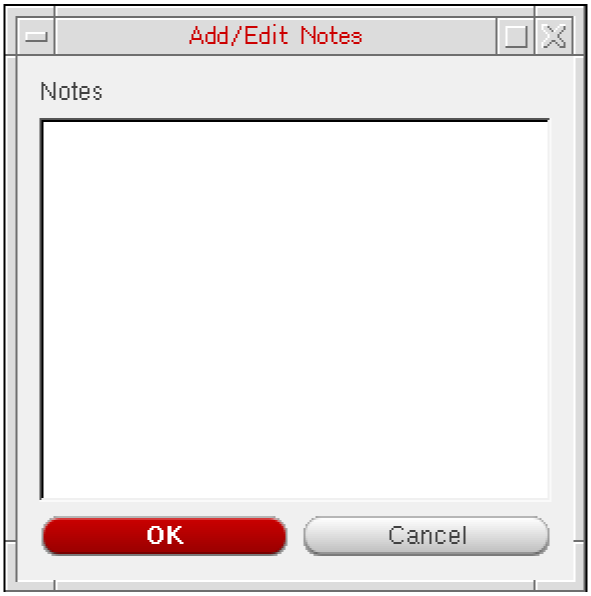
- ➡ On the Data View pane, right-click the corner name and choose *Rename*. The name of the corner becomes editable. Change the name of the corner and press *Enter*.
- ➡ In the Corners Setup form, select the name of the corner. The name of the corner is displayed in *Corner Name* text field. Change the name of the corner and press *Enter*.

Adding Notes to a Corner

If you have some additional information or notes to be added to the corners so that it can be referred to later, you can add notes for each individual corner. To add notes for a corner, do the following on the Data View pane or in the Corners Setup form:

1. Right-click the corner name and choose *Notes*.

The Add/Edit Notes form is displayed.



2. In the *Notes* field, add notes for the test.
3. Click *OK*.

The notes are saved in the setup database and displayed in the tooltip for the corner.

For related information, see Adding Notes to a Test.

Disabling and Enabling Corners

By default, a corner is enabled to run for all the tests in the current ADE XL setup. You can selectively disable and enable the nominal corner, one or more corners for one or more tests, or all corners.

Disabling and Enabling the Nominal Corner

To disable/ enable the nominal corner simulation for specific tests, deselect/select the check box next to a test name in the column for the nominal corner.

To disable the nominal corner simulation for all tests, do one of the following:

➡ On the Run Summary assistant pane, deselect the *Nominal Corner* check box.

➡ In the Corners Setup form, deselect the *Nominal* check box.

➡ In the Corners tree on the Data View pane, deselect the *Nominal* check box.

Similarly, to enable the nominal corner simulation for all tests, select the *Nominal* check box at any of the three locations listed above.

Disabling and Enabling All Corners for Specific Tests

To disable all corners for a test, do the following in the Corners Setup form:

➤ Right-click on a test name and choose *Disable Corners on Test*.

The check boxes for all the corners for the test are selected. The simulator does not run the corner simulations for the test.

To enable all corners for a test, right-click on a test name and choose *Enable Corners on Test*.

Disabling and Enabling One or More Corners for All Tests

To disable one or more corners for all tests, do the following in the Corners Setup form:

1. Select a corner.
2. (Optional) Hold down the *Shift* key (for contiguous corners) or the *Ctrl* key (for noncontiguous corners) and click the next corner to add more corners to the selection set.
3. Right-click the name of a corner and choose *Disable*.

   The check boxes next to the names of all corners in the selection set are deselected. The simulator does not run those corner simulations.

To enable one or more corners for all tests, select the name of the corners as described above, and then right-click the name of a selected corner and choose *Enable*.

Disabling and Enabling One or More Tests for One or More Corners

To disable one or more tests for one or more corners, do the following:

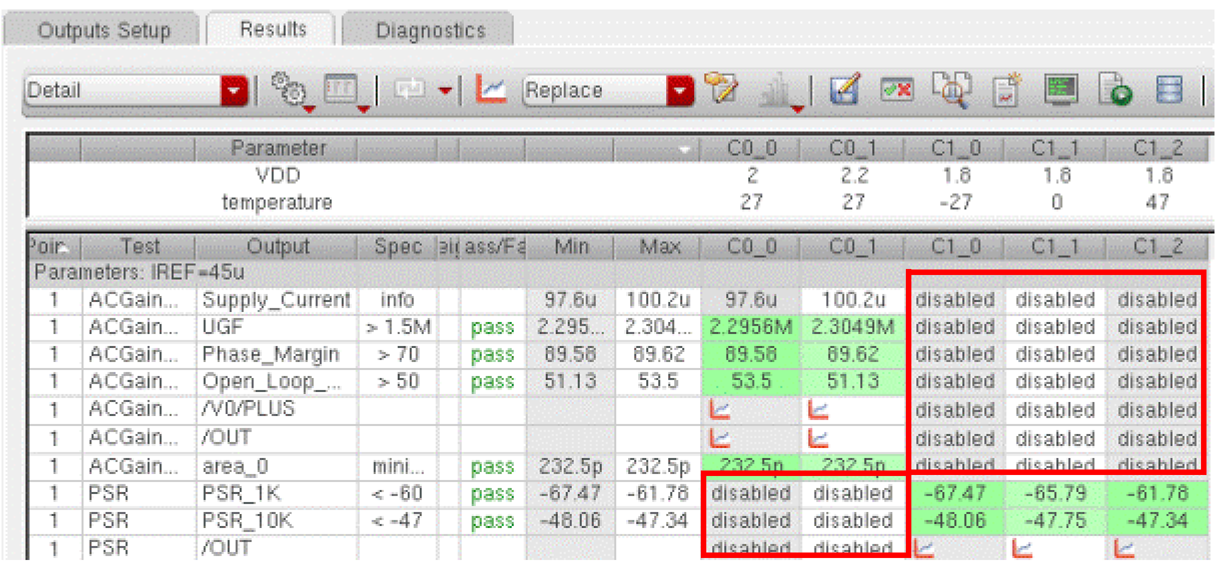
➡ In the Corners Setup form, deselect the check box next to the name of the test in the column for a corner.

To disable a test for multiple corners, select all the required corners and deselect the check box next to the name of the test for one corner. The check boxes for all the tests in the selection set are deselected. The simulator does not run corner simulations for these tests.

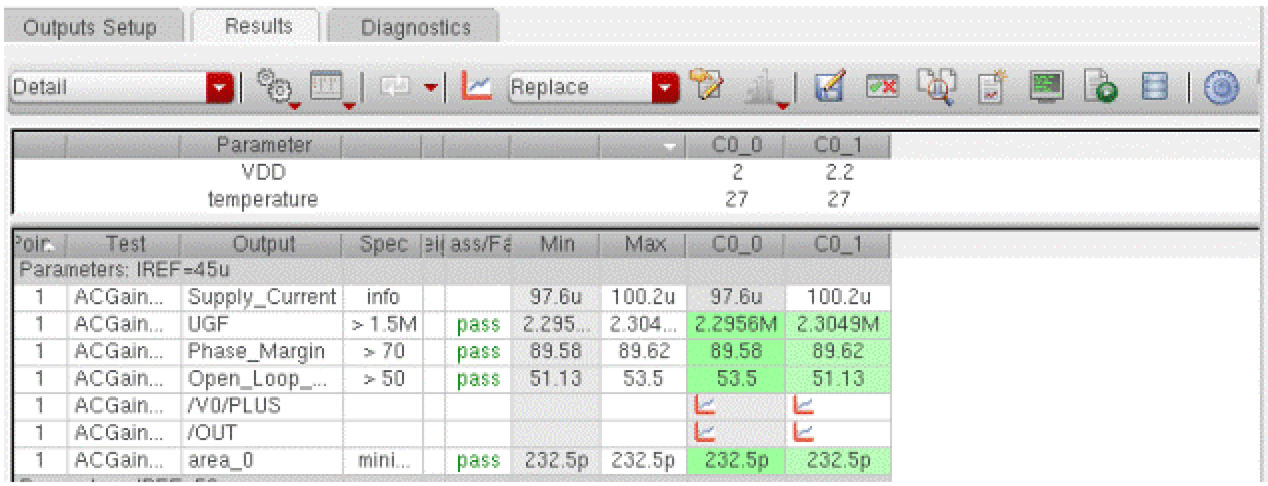
To enable one or more tests for one or more corners, select all the required corners and select the check box next to the name of the test for one corner. The program selects the check boxes for all the tests in the selection set. The simulator runs corner simulations for these tests.

Viewing Disabled Corners in the Results Tab

When a corner is disabled for a test, the Results tab shows `disabled` in the cells corresponding to that corner-test combination, as shown in the figure below.



Results in the above example show that corner `C1` is disabled for the test `ACGain` and corner `C0` is disabled for `PSR`. If you hide or filter out the results for either one of these two tests, the columns for the corners that are disabled for the other test are automatically hidden. The following figure shows how the results appear when the `PSR` test is hidden.



Note that the three columns for corner `C1` are now hidden because that corner is disabled for `ACGain`.

Also see: Hiding and Showing Results for Tests

Disabling and Enabling All Corners

To disable all corners, do one of following:

- In the Run Summary assistant pane, clear the *# Corners* check box (where *#* indicates the number of enabled corners).

  The *Nominal Corner* check box also becomes unavailable. The simulator does not run any corners.

- In the Data View pane, clear the check box next to the *Corners* tree.

To enable all corners, select the check box at any of the two locations specified above.

Removing Corners

To remove one or more corners, do the following in the Corners Setup form:

1. Select the name of the corner you want to remove.

2. (Optional) Hold down the *Shift* key (for contiguous corners) or the *Ctrl* key (for noncontiguous corners) and click the next corner to add more corners to the selection set.
3. Click the *Remove Corner*  button.

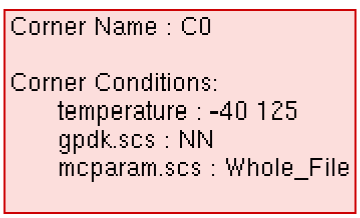
   The selected corner columns disappear from the form. The number of *Corners* reflected on the Run Summary assistant pane also decreases by the number of corners you remove.

Viewing Corner Settings

To view the settings for a corner, do the following:

➤ In the Corners tree on the Data View pane, hover the mouse pointer over the name of a corner.

The corner settings are displayed in a pop-up, as shown in the following figure:



Copying Corners

You can reuse the information in a corner by creating a copy of the corner.

To create a copy of a corner, do the following in the Corners Setup form:

1. Right-click the corner name and choose *Create Corner Copy*.

   A new column containing the same corner information is displayed in the Corners Setup form.

Note: You can also create copy of an expanded corner based on its results in the Results tab. For more details, refer to Simulating Corner on page 262.

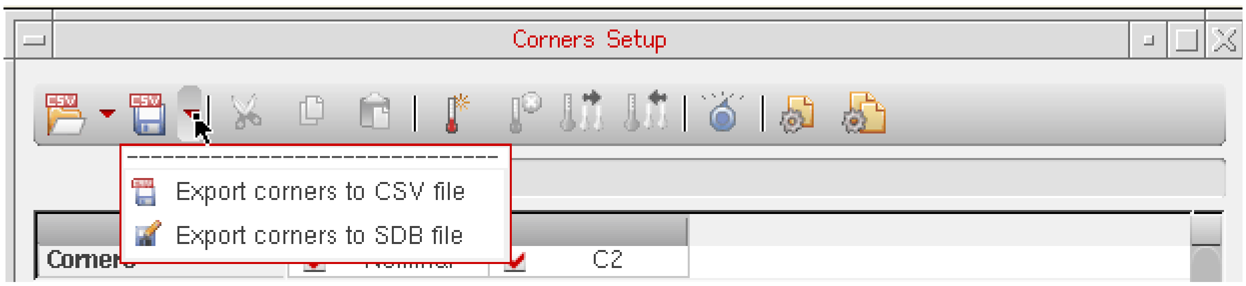
Exporting Corners

You can export corners from the Corners Setup form and save them in the following formats:

- XML format in a .sdb file (setup database)
- CSV format in a .csv file

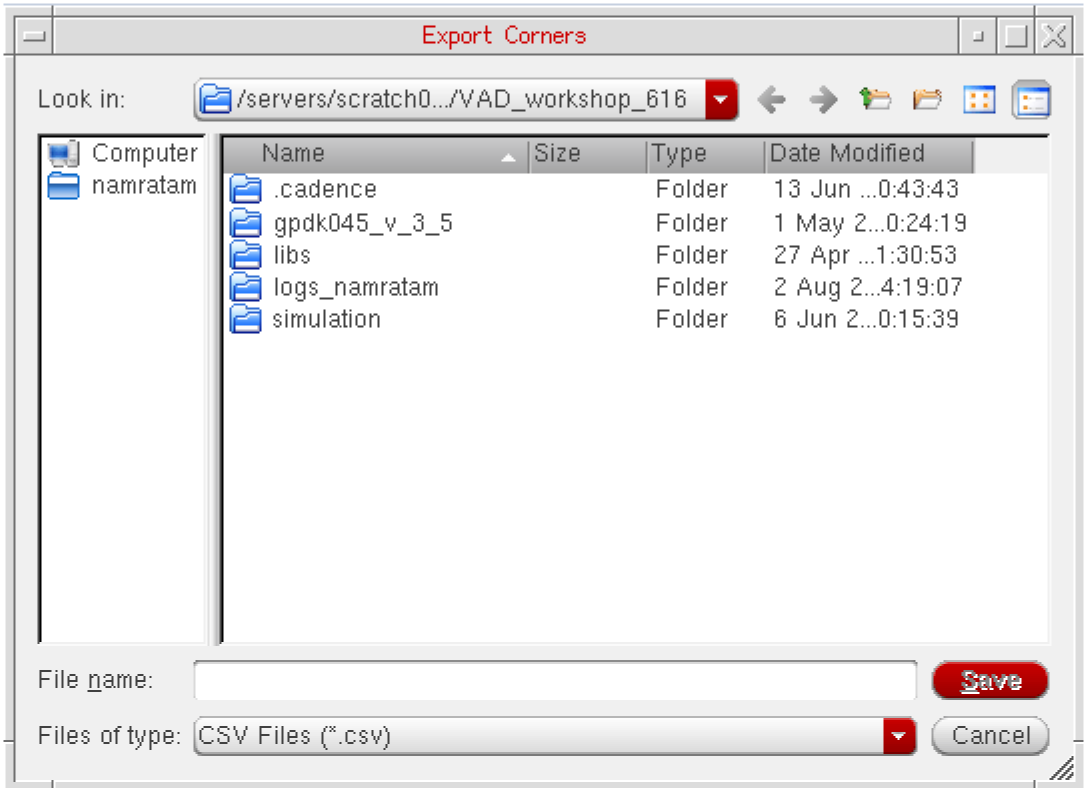
To export corners setup information in any of these formats, do the following:

1. In the Corners Setup form, choose an appropriate command to export the corners, as shown in the figure below.



Note: The default button displayed in the drop-down menu is defined by the defaultCornerExportFileFormat environment variable. If required, you can change the default command by setting the value of this environment variable before opening the Corners Setup form.

The Export Corners form is displayed, as shown below.



Note: The file extension displayed in the *Files of type* drop-down list box is determined by the export command that you select. If required, you can change the file type.

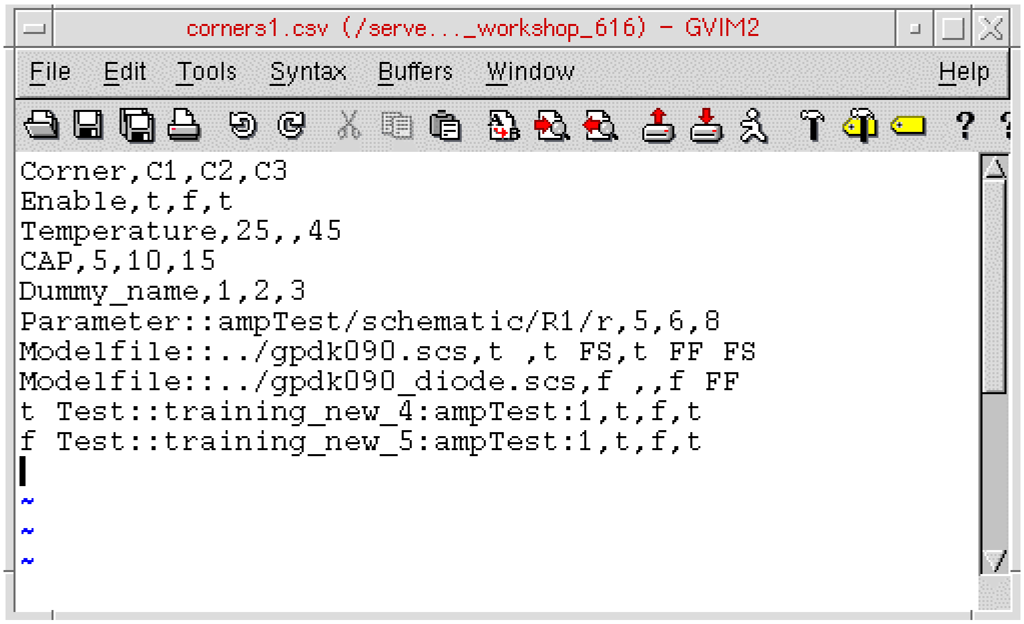
2. (Optional) Browse and select a folder in which you want to save the file.

3. In the *File name* field, type a name for the file in which you want to save the exported corners setup.

4. Click *Save*.

The corners setup information is saved in the specified `.csv` or `.sdb` file.

If you saved the corners in the `.csv` format, you can open the file to view or modify the details. However, you need to ensure that the format of the file is correct. For details on the format of a CSV file, refer to Format Details of a Corners CSV File.



Important

When you export the corner details in the CSV format, ADE XL ignores the following setup details:

- Model groups specified for any corner
- Statistical corners
- Worst case corners

If you have defined any of these details in the Corners Setup form, ADE XL does not save these in the `.csv` file and displays an appropriate message.

Importing Corners

You can import the corner details from an existing .csv or .sdb or a customization file.

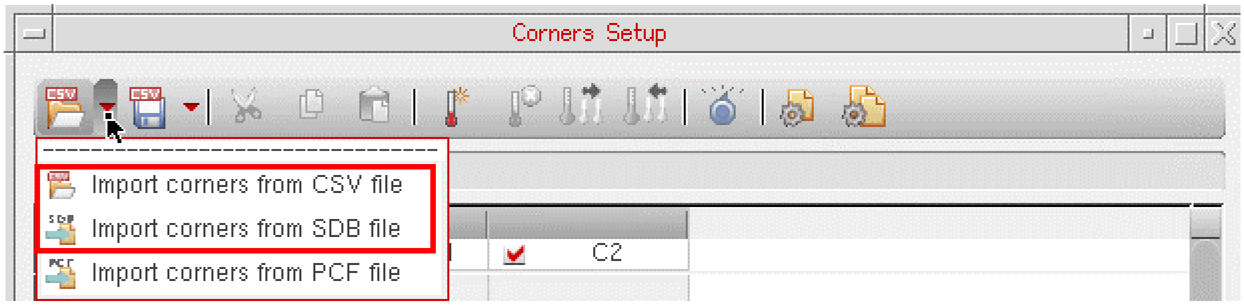
The following topics explain how to import corners from these files:

- Importing Corners from .csv or .sdb Files
- Importing Corners from Customization Files

Importing Corners from .csv or .sdb Files

To import the corners setup from a .csv or .sdb file, do the following:

1. In the Corners Setup form, choose an appropriate command to import the corners.

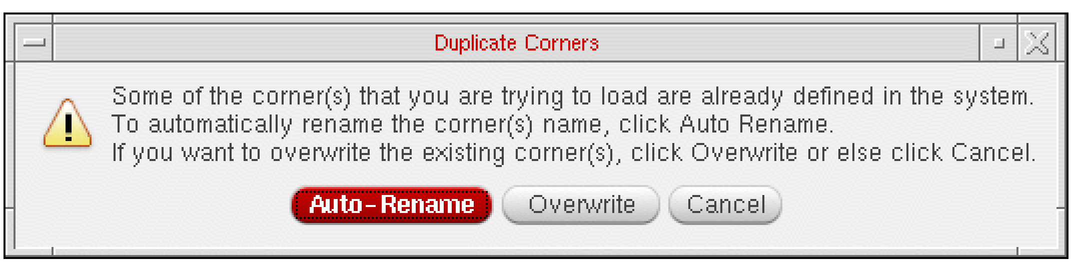


Note: The default button displayed in the drop-down menu is defined by the defaultCornerImportFileFormat environment variable. If required, you can change the default command by setting the value of this environment variable before opening the Corners Setup form.

The Import Corners form is displayed.

2. Browse and select a valid corners setup file.
3. Click *Open*.

The corners setup information is loaded from the specified file and displayed in the Corners Setup form. If the file has a corner that has the same name as an existing corner, the following message is displayed.



Click *Auto-Rename* to import the duplicate corner from the file with a different name. For example, if there is an existing corner named `C1`, the corner in the file will be imported as `C1_0`. In this case, both the existing corner and the corner being imported from the file are used.

Click *Overwrite* to overwrite the existing corner with the corner being imported. In this case, the column for the existing corner is removed from the corners setup table.

Click *Cancel* to cancel the import process.

Important

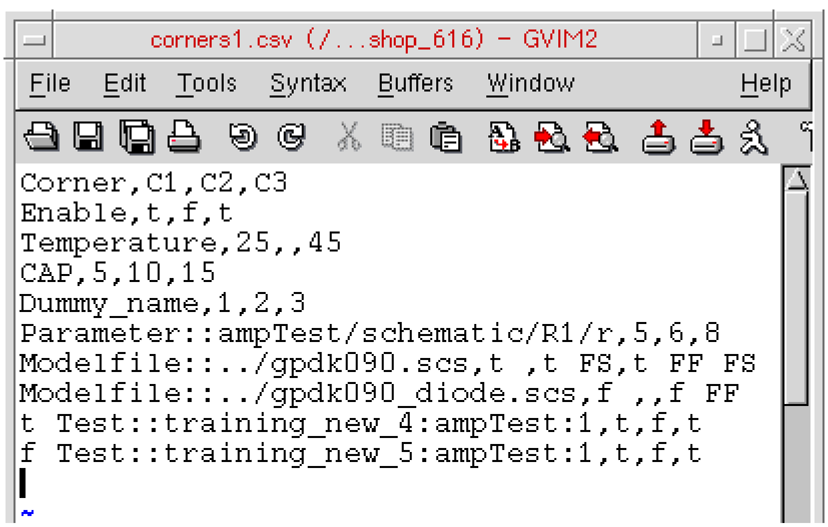
When you import the corner details from a `.csv` file, ADE XL ignores the global variables and parameters that are not found in the list of global variables and parameters in the Data View pane. A corresponding warning message is displayed in the CIW.

Format Details of a Corners CSV File

While importing corner details from a CSV file, ADE XL requires the information to be given in a desired format that it uses to identify different elements of a corner. This section explains how each row in a CSV file is required to be formatted so that the information can be imported by ADE XL into the Corners Setup form.

The figure given below shows a sample CSV file.

Line 1
Line 2
Line 3
Line 4
Line 5
Line 6
Line 7
Line 8
Line 9
Line 10



In the above figure, the lines have been numbered so that they can be referenced to explain the format.

Line#1: The names of corners are identified by a comma-separated list given in a line that begins with the keyword `Corner`.

Line#2: The state of a corner, enabled or not, is identified by a line that begins with the keyword `Enable`. In the example, the corner `C2` is not enabled. If a blank entry is found in this list, ADE XL considers the default value and marks the corresponding corner as enabled.

Line#3: The temperature value for each corner is defined by a comma-separated list given in a line that begins with the keyword `Temperature`. A blank entry in this list is considered as a `NULL` value for the corresponding corner.

Line#4, 5: The lines that define the design variables and their values for each corner begin with variable names. There can be many lines for the design variables where each line is for a different variable. If more than one line is given for a single variable, only the values given in the last occurrence are used. A blank entry in the list for a variable is considered as a `NULL` value for the corresponding corner. If a variable name does not match with any variable for the tests in the setup, it is ignored and its details are not imported in the form. An appropriate warning message is also displayed in the CIW. In the example shown above, `Dummy_name` would be ignored.

Line#6: The lines that begin with the keyword `Parameter::` define the parameters and their values for each corner. A separate line is given for each parameter. If a parameter name does not exist in the *Parameters* pull-down list in the Corners Setup form, it is ignored and not imported. An appropriate warning message is also displayed in the CIW.

Note: A comma-separated list of values assigned to a single corner column for `Temperature`, design variable, or parameter is to be enclosed in double-quotes, as shown in the following example:

```
Corner,C0,C1,C2,C3
Enable,t,t,f,t
Temperature,"40,75",25,,45
vdd,2.0,"2.2,2.4",, 2.0
```

In this case, value of `Temperature` for corner `C0` is `40,75`.

Line#7, 8: The lines that begin with the keyword `Modelfile::` define the model file names, their enabled status and the section names to be used for each corner. Separate rows are used for separate model files. Each value in this list is a space-separated sublist that gives the enabled status of the model for a corner and then the list of section names. The right-most value in line#7, `t FF FS`, mentions that the model file has been enabled for corner `C3` and sections `FF` and `FS` are to be used for `C3`.

When a blank entry is found in this list, the model file is not used for the corresponding corner. In the above example, the `gpdk090_diode` model file is not used for corner `C2`.

Note: If the section names are not given and the model file entry only shows the enabled status for a corner, the correct format is `<t` or `f><space>`. If only `t` or `f` is given, it is considered as a section name to be used from the given model file.

Line#9, 10: The usage of corners for tests is defined by the rows that begin with the keywords `t` or `f` followed by the keyword `Test::`, the test name, and the list of enabled status for each corner. In the following line from the example CVS file:

`t Test::training_new_4:ampTest:1,t,f,t`

`t` in the beginning states that corners will be run for the `Test::training_new_4:ampTest:1` test. The `t,f,t` list states that corner `C2` is not enabled for this test.

Note: Any statement that does not begin with a recognized keyword is considered as an entry for a design variable. If the first word is found in the list of available design variables, it is imported into the Corners Setup form. Otherwise, a warning message is displayed in the CIW.

Importing Corners from Customization Files

You can import a set of predefined corners and their measurements from the following types of customization files:

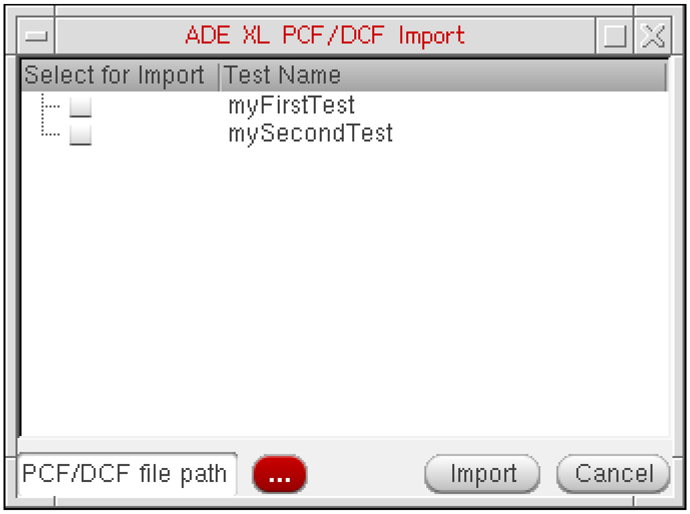
- Process Customization Files (PCFs): PCF is a file that is typically created by a process engineer or process group, and defines the processes, groups, variants, and corners so that everyone in an organization uses the same definitions.
- Design Customization Files (DCFs): DCF is a file that is typically created by a design engineer or design group ,and defines the corner variables and measurement information for a particular design or for several designs within a design group.

Usually, you need to load the process customization information before you can import corners or measurements from a design customization file. The settings must refer to the definitions that you have already loaded.

For each customization file that you want to import, do the following:

1. In the Corners Setup form, click *Import Corners from PCF File* () on the toolbar.

The ADE XL PCF/DCF Import form is displayed.



2. Click the browse button.

   The Choose PCF/DCF File for Import form is displayed.

3. Browse and select the PCF/DCF file from where you want to import the corners.

4. Click *Open*.

   The file is displayed in the field in the lower left corner of this form.

5. Select the check boxes next to each test name for which you want to use the imported corners.

6. Click *Import*.

   The program imports the corners and their measurements from the specified file. The new corners are added as new columns in the Corners Setup form. If required, you can edit the corner details, such as variable values or model sections. When you edit the section name for a model file, simulation for the corner would fail if the specified section name does not exist in the limited list specified in the PCF file. To avoid this, you can set the LimitModelSections environment variable to `LimitedList` to specify that if the section name does not exist in the limited list, the tool should display an error message in the Corners Setup form.

   Note the following:

   - If the program finds any errors while reading the file, these messages appear in the CIW output area.

- If more than one process has a corner set with the same name, the corners will be displayed as `processName_cornerName` in the Corners Setup form. For example, if the process named `myProcess` defines a corner set named `sample` and another process named `myProcess1` also defines a corner set named `sample`, the corners are displayed as `myProcess_sample` and `myProcess1_sample` in the Corners Setup form.

Setting Up a Default Set of Corners

Do the following to automatically setup a default set of corners when you open an ADE XL view:

1. Use the procedure described in Exporting Corners on page 247 to create a file that contains a set of corners that you want to be used as the default set of corners in every ADE XL view.
2. Use the `adexl.gui defaultCorners` environment variable to automatically load the default set of corners from the file when you open an ADE XL view.

   You can use the default set of corners as is, or use them to quickly setup up corners for your tests.

Important

The default corners will be loaded only if no other corner is defined in the Corners Setup form in the ADE XL view.

The following examples describe the use of the `adexl.gui defaultCorners` environment variable.

Example 6-1 Loading a default set of corners using the defaultCorners environment variable

Add the following entry in your `.cdsenv` file to load the corners defined in the `myDefaultCorners.sdb` file:

```
adexl.gui defaultCorners string "$HOME/myDefaultCorners.sdb"
```

Example 6-2 Loading a default set of corners based on a custom SKILL procedure

Add the following procedure in your `.cdsenv` file to load a default set of corners defined in the `myDefaultCorners.sdb` file based on the `myTechProp` property assigned to a cellview in a technology library named `myTechLibs`:

```
procedure(myCornerLoadProc(args)
     session = asiGetSession(args->window)
     myTechName = ddGetObj(asiGetTopCellView(session)->libName)->technology
     envSetVal( "adexl.gui" "defaultCorners" 'string strcat("/cadence/virtuoso/"
     myTechName "/libraries/myTechLibs/" getShellEnvVar("myTechProp")
     "/myDefaultCorners.sdb"))
)
deRegUserTriggers("adexl" nil nil 'myCornerLoadProc)
```

Working with Corner Groups

A corner group is a set of corners that ADE XL displays as a single corner. For more information about working with corner groups, see the following topics:

- Creating a Corner Group on page 257
- Expanding a Corner Group on page 260

Creating a Corner Group

When two or more corners have same corner settings except few, you can group all these corners to create a corner group. For example, in the following figure, four corners, C1, C2, C3, and C4 have same values for Temperature, param1, and Model Group. However, these corners use different corner values for the design variable `vdd` and different sections of the same model file.

Corners	✔ Nominal	✔ C1	✔ C2	✔ C3	✔ C4
Temperature		10	10	10	10
Design Variables					
vdd		12	12	10	12
Click to add					
Parameters					
...RR_SR/I59/param1		10	10	10	10
Click to add					
Model Files					
gpdk090.scs		✔ FF	✔ SS	✔ FS	✔ FS
Click to add					
Model Group(s)		modelGrp1	modelGrp1	modelGrp1	modelGrp1
Click to add					
Tests					
✔ myFirstTest	✔	✔	✔	✔	✔
✔ mySecondTest	✔	✔	✔	✔	✔
Number of Corners	1	1	1	1	1

You can create groups for these corners so that the Corners Setup form displays these corners in a single column. This helps in a better management of these corners.

You can create corner groups using the Corners Setup form or the Data View pane.

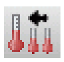
Creating Corner Groups Using the Corners Setup Form

To create a corner group using the Corners Setup form, do one of the following:

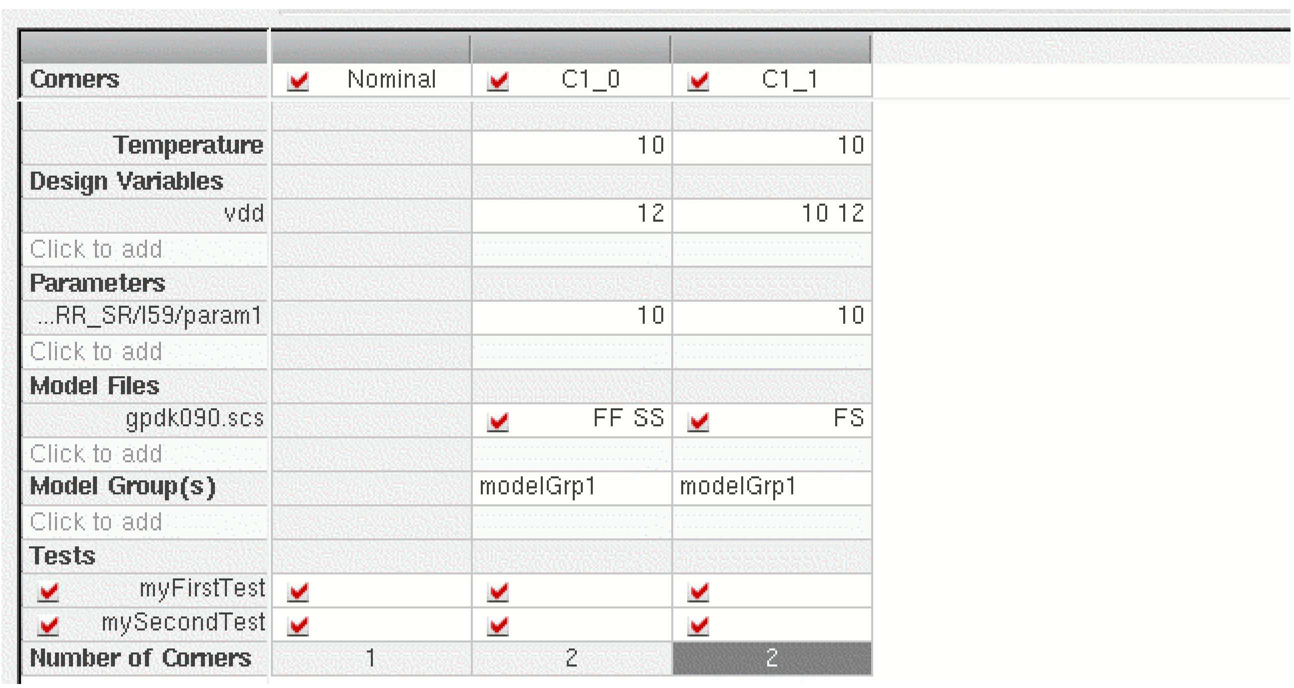
- Select the names of the corners you want to include in the corner group by doing the following:

    a. Click a corner name.

    b. Hold down the *Shift* key (for contiguous corners) or the *Ctrl* key (for noncontiguous corners) and click the next corner name to add more corners to the selection set.

    c. Click .

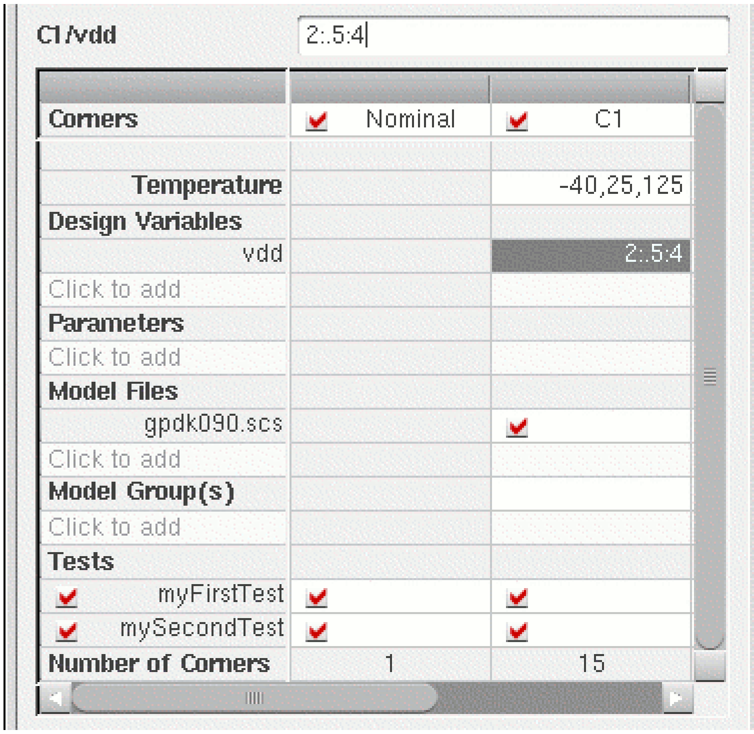
    The corners are combined into one or more corner groups depending on the corners settings. For the example shown above, the program creates two groups, as shown in the following figure.



Note that two corner groups, C1_0 and C1_1 have been created. The *Number of Corners* field for both the groups shows the value `2`. This implies that both the groups contain two corners each. This also indicates that for each corner group, two simulations will run.

Another way in which you can create a corner group in the Corners Setup form is by adding multiple values for corners settings.

For example, for the `C1` corner in the following figure, a list of values (`-40,25,125`) is specified for the `temperature` parameter, and a range of values (`2:.5:4`) is specified for the `vdd` design variable.



The program calculates creates all possible combinations using these lists or ranges of corner values and writes the number of corners that will run for the corner group.

Creating Corner Groups Using the Data View Pane

To create a corner group using the Data View pane, do the following:

1. In the Corners tree on the Data View pane, select the names of the corners you want to include in the corner group.

   Hold down the *Shift* key (for contiguous corners) or the *Ctrl* key (for noncontiguous corners) and click the next corner name to add more corners to the selection set.

2. Right-click and choose *Create Corner Group*(s).

   The corners are combined into a single corner group. For example, if you group corners named *C1*, *C2* and *C5*, a corner group named *C1_0* is created.

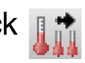
Expanding a Corner Group

This section describes, how to:

- Expand a corner group for all possible combinations of corner parameters
- Expand a corner group for selected corner parameters
- Undo an expanded corner group

Expanding Corner Group for all Combinations of Parameters

To expand a corner group so that it is displayed as several individual corners, do one of the following:

- In the Corners Setup form, select a corner group name and click .
- In the Corners tree on the Data View pane, right-click a corner group and choose *Expand Corner*.

The program expands the corner group and creates corners for all possible combinations of corner values. Each expanded corner is named from the original corner followed by and underscore and a sequence number starting at zero.

For example, if you expand a corner group that contains two variables such as *Temperature*=`-40,27,125` and *vdd*=`2,2.5,3`, the total number of expanded corners for these values is nine. If the original corner name is *C5* (for example), the expanded corner names are *C5_0* through *C5_8*.

Expanding Corner Group for Selected Parameters

If there are many corner variables, you may want to expand a corner by using values of only one or more parameters. To expand corner groups for selected variables:

1. Press and hold the `Ctrl` key and click on the cells of selected parameters of the corner group.
2. Right-click and choose *Expand Corner Selectively – Selected Params*.

The corner group is expanded for all combinations of values of the selected parameters.

> **Note:** To expand corners for value sets of the selected parameters, choose *Expand Corner Selectively – Selected ParamSet*. For example, in a corner group, `C_temp_VDD`, *`temp`* is varied for three values: `-25`, `40` and `75`. *`VDD`* is also varied for three values: `1.8`, `2.0`, and `2.4`. If you choose the *Expand Corner Selectively –*

Selected ParamSet command, group1 will be expanded to create the following three corners:
- C_temp_VDD_0 with `temp` = `-25` and `VDD` = `1.8`
- C_temp_VDD_1 with `temp` = `40` and `VDD` = `2.0`
- C_temp_VDD_2 with `temp` = `75` and `VDD` = `2.4`

Undo Expanded Corner Group

To undo the expand of a corner group, do one of the following:

- In the Corners Setup form, right-click an expanded corner name and choose *Undo Expand Corner*.
- In the Corners tree on the Data View pane, right-click an expanded corner name and choose *Undo Expand Corner*.

Important

After expanding a corner group, if you modify values of the corner variables, the *Undo Expand Corner* command will not be available. This is because now you need to recreate corner groups based on the new set of corner values.

Simulating Corner

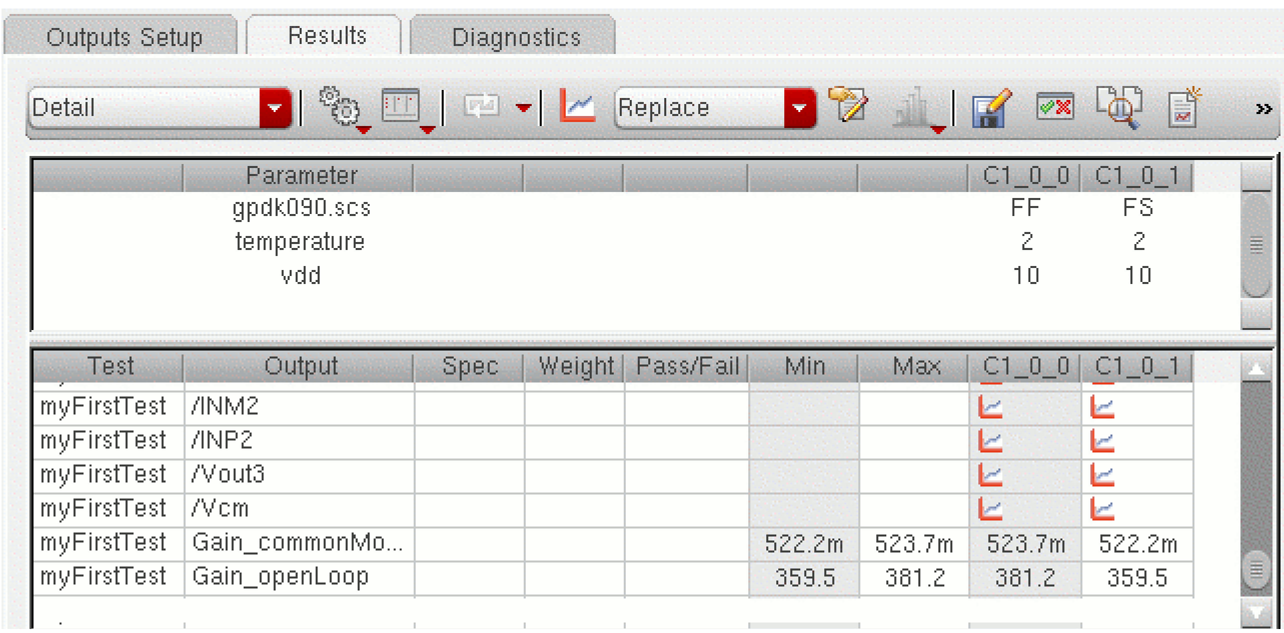
To run a simulation for a specific corner, do the following:

➡ In the Corners tree on the Data View pane, right-click on a corner and choose *Run*.

To run a simulation for one or more corners, do the following:

1. In the Data View pane, ensure that the name of the corners for which simulation is required to be run are selected in the Corners tree.
2. Click *Run Simulation*  on the Run toolbar.

The simulator runs simulation for the selected corners and shows results on the *Results* tab. If you run a corner group, simulations are run for all the corners in that group and results are displayed as shown in the following figure.



Note that in case of a corner group, a column is displayed in the Results tab for each combination of corner parameters. Based on the results, if you want to create a corner by using the values of variables used for a particular data point, you can create a copy of that corner. For this, right-click in the corner column and choose *Create Copy of Selected Corner*. A new corner is created with the name same as that of the corner column, prefixed with `Copy_`.

Important

While running simulations with corners, ADE XL sorts the values of variables and model sections in an alphabetical order. It also saves and displays the results in the sorted order. To disable sorting of variable values and model sections and to use the variables in the specified order, set the sortVariableValues variable to `nil`.

7

Performing Monte Carlo Analysis

You can use Monte Carlo (statistical) analysis to estimate parametric yields and generate information about the performance characteristics of the circuits you design.

The manufacturing variations in components affect the production yield of any design that includes them. Statistical analysis allows you to study this relationship in detail.

To prepare for a statistical analysis, you create a design that includes devices or device models that are assigned statistically varying parameter values. The shape of each statistical distribution represents the manufacturing tolerances on a device. During the analysis, the statistical analysis option performs several simulations. Each simulation uses different parameter values for the devices based on the statistical distributions you specify.

When the simulations finish, you can examine how manufacturing tolerances affect the overall production yield of your design. If necessary, you can use different components or change the design to improve yield.

For more information, see the following topics:

- Running a Monte Carlo Analysis on page 266
- Including or Excluding Instances and Devices for Applying Mismatch Variations on page 276
- Stopping Monte Carlo Based on the Target Yield on page 293
- Viewing Monte Carlo Results on page 297
- Managing Monte Carlo Results in the Yield View on page 298
- Creating Statistical Corners on page 299
- Generating Plots, Tables, and Reports on page 310
- Viewing Sensitivity Results on page 327
- Viewing Statistical Parameters for Monte Carlo Samples on page 327
- Running Multi-Technology Simulations for Monte Carlo Analysis on page 328

Running a Monte Carlo Analysis

Important

Monte Carlo analysis is available only for the following simulators:

- Virtuoso Spectre circuit simulator
- Virtuoso Accelerated Parallel simulator (APS)
- Virtuoso AMS Designer simulator with Spectre or APS as the solver

  To select a solver, in the Data View pane, right-click on the test name and choose *High-Performance Simulation*. Select a solver in the High-Performance Simulation Options form.

  Note the following:

  - Ensure that you are using the Cadence IUS 9.2 or later version of Virtuoso AMS Designer simulator.
  - You cannot use Virtuoso AMS Designer simulator with UltraSim as the solver to run Monte Carlo analysis.
  - Running Monte Carlo simulation in interactive mode (using SimVision) is not supported with the AMS Designer simulator.

You can run Monte Carlo analysis over more than one test and corner. Your design must include devices or device models for which you have specified statistically varying parameter values. You must have one or more specs defined and enabled. You must specify either global (process) or mismatch (per-instance) variations or both. You can also specify correlation information by specifying correlation constraints in Constraint Manager. After simulating, you can select the yield view to view mean and standard deviation information.

Note: For information about specifying parameter distributions for Spectre circuit simulation, see Mone Carlo in the *Virtuoso Spectre Circuit Simulator and Accelerated Parallel Simulator User Guide*.

To run Monte Carlo analysis over more than one test or corner, do the following:

1. On the Data View assistant pane, identify the tests over which you want to run Monte Carlo by selecting their check boxes.

   The Run Summary pane reflects the number of tests you have selected.

2. Ensure that the simulator specified for the tests is Spectre, APS, or AMS Designer (with Spectre as the solver). You can run Monte Carlo analysis only with these simulators.

3. Disabled output expressions—expressions for which the *Plot* check box is not selected in the Outputs Setup tab—will not be evaluated for Monte Carlo simulations. So ensure that the *Plot* check box is selected for the output expressions that you want to be evaluated.

4. (Optional) Specify correlation constraints in Constraint Manager. The correlation constraints are evaluated during the Monte Carlo run. For more information, see Chapter 5, "Working with Constraints."

5. (Optional) If you have corners defined, you can also select the *# Corners* check box on the Run Summary pane to run Monte Carlo over them.

   Note: You can change the corners you have defined on the Corners Setup form.

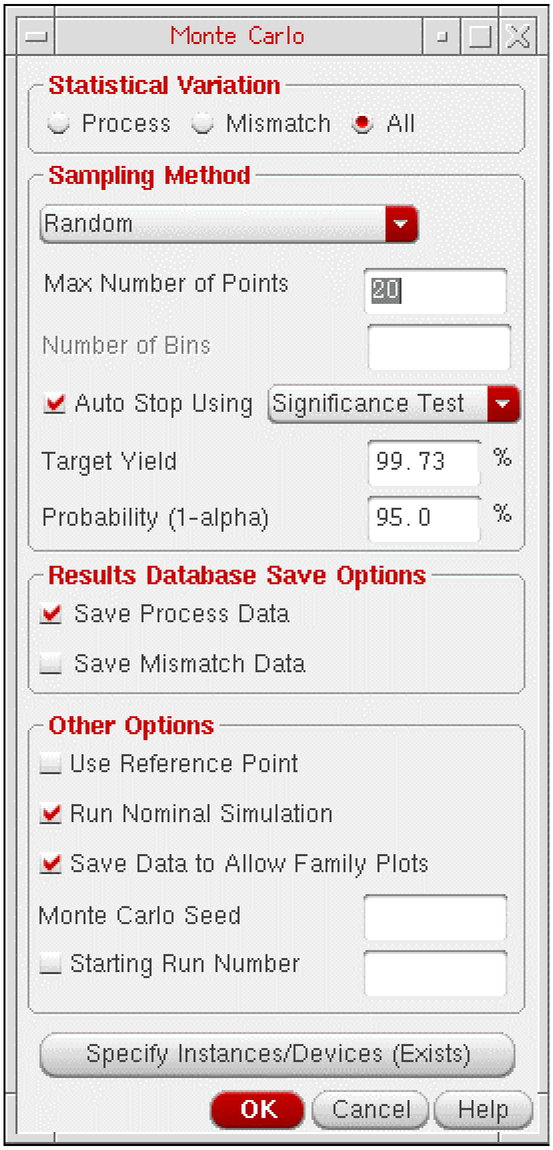
6. (Optional) If you do not want to include the nominal corner in the Monte Carlo analysis, you can deselect the *Nominal Corner* check box on the Run Summary pane.

7. From the *Run* menu, select *Monte Carlo Sampling*.

Tip

Alternatively, select *Monte Carlo Sampling* in the *Select a Run Mode* drop-down list on the Run toolbar, then click the *Simulation Options* button on the Run toolbar.

The Monte Carlo form appears.



8. In the *Method* group box, select one of the following statistical variations:

Process	for process statistical variations
Mismatch	for per-instance statistical variations
All	for both process and per-instance statistical variations

Important

You must define your models so that they respond to the statistical variations you choose. You must specify the file containing your models on the Model Library Setup form. For a Spectre circuit simulator example of how to define your models, see "Specifying Parameter Distributions Using Statistics Blocks" in the *Virtuoso Spectre Circuit Simulator User Guide*.

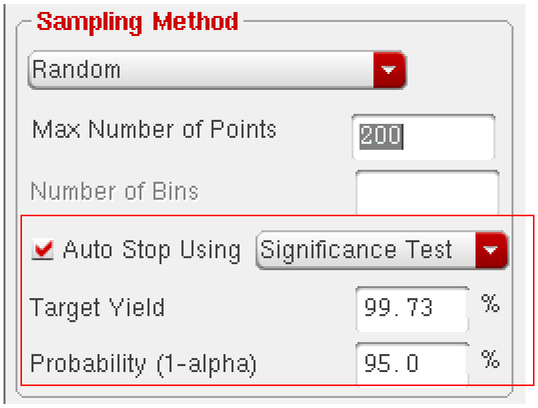
9. In the *Number of Points* field, type the number of Monte Carlo points you want to simulate.

10. In the *Sampling Method* group box, select any of the following statistical sampling method to be used:

- `Random`*:* The Random sampling method takes the Brute Force approach of sequentially calling a random number generator without considering the samples generated previously. There is no sample selection or rejection. It has a convergence accuracy of 1/sqrt(N).
- `Latin Hypercube`: Latin Hypercube is a quasi-random sampling algorithm with sample selection and rejection. The sample space is evenly divided into probable subspaces. All sample points are then chosen simultaneously making sure that the total ensemble of sample points is a Latin Hypercube sample and that each subspace is sampled with the same density. The Latin Hypercube algorithm has a convergence accuracy of 1/pow(N,2/3).
- `Low-Discrepancy Sequence`: Low-Discrepancy Sequence (LDS) uses a deterministic sequence to get a uniform coverage of the sampling space, which makes it better than the `Random` sampling method. In addition, `LDS` uses auto stop features to generate samples, which is not supported by `Latin Hypercube Sampling (LHS)`. The convergence speed for `LDS` is faster than the `Random` sampling method and is comparable to the `LHS` method. Therefore, overall, this method is better than both, `Random` and `LHS`.

  If the selected sampling method is `Random` or `Low-Discrepancy Sequence`, you can choose to automatically stop the simulation based on the following two criteria:

  - Auto Stop Using Significance Test: This option applies a stopping criteria based on the target yield value. To enable this stopping criteria, select `Significance Test` from the *Auto Stop Using* drop-down list, as shown below.

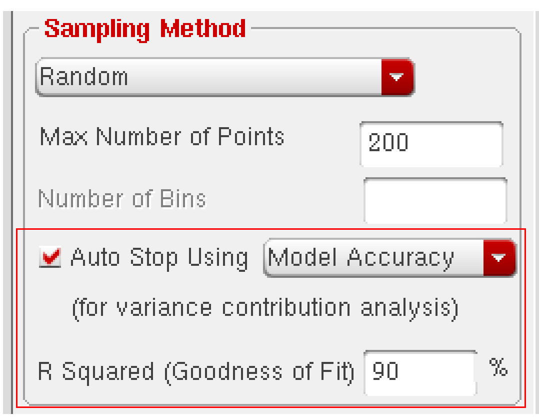


For more details on how a significance test helps in stopping the Monte Carlo run, refer to Stopping Monte Carlo Based on Target Yield.

- Auto Stop Using Model Accuracy: This option applies a stopping criteria based on the accuracy of the modeling of variation in the outputs, which is due to statistical variation.

  Note: Enable this option when you do not know how many samples are required to view mismatch contribution results.

  To enable this stopping criteria, select `Model Accuracy` from the *Auto Stop Using* drop-down list and specify a target accuracy level in the *R Squared (Goodness to Fit)* field, as shown below.



  When this auto stop option is used, the Monte Carlo run is stopped as soon as the required number of samples to achieve the target accuracy level are run. The target accuracy level is set to `90%` by default, which means the Monte Carlo analysis will stop after simulating enough samples that will build a model that explains 90% of the variance in all output measurements. A higher value of the accuracy level ensures more effective modeling of data.

Note: The modeling of variation in the outputs due to statistical variation can be done by using the post processing option to view and analyze the mismatch contribution results. Therefore, when using this option, it is required that you choose to run mismatch statistical variation and also select the *Save Mismatch Data* option to save the data for Monte Carlo analysis so that it can be later used for post-processing.

It is not essential to set a target to automatically stop the Monte Carlo analysis because you can set the number of points in the *Max Number of Points* field. However, it is recommended to use this option when you are not sure about the number of samples to run. This option helps in automatically determining the number of samples required to run to meet the target accuracy level so that post-processing for mismatch contribution analysis can be done. The number of samples depends on the design and the number of statistical parameters defined by the model files.

Note: You can use the maxOutstandingPoints environment variable to set the maximum number of outstanding points to be submitted at a time before determining whether to stop a Monte Carlo run.

For more details on the use of this option for variance contribution analysis, refer to Analyzing the Mismatch Contribution in the *Analog Design Environment GXL User Guide*.

11. If the selected sampling method is `Latin Hypercube`, specify the number of bins (subdivisions) in the *Number of Bins* field.

Note the following:

- ❑ If a number is specified, the number of bins must be greater than the *Number of Points* + *Starting Run Number* - `1`. If this condition is not satisfied, a warning message is displayed and you are suggested to change the values so that the required condition is met.
- ❑ If a number is specified, the number of bins will be the specified number or *Number of Points* + *Starting Run Number* - `1`, whichever is greater. For example, if the specified number of bins is `90`, the number of points specified in the Number of Points field is `100` and the starting run number specified in the Starting Run Number field is 6, the value `105 (100+6-1)` is used.
- ❑ If no number is specified, a default value of *Number of Points* + *Starting Run Number* - `1` is used. For example, if the number of points specified in the *Number of Points* field is `100` and the starting run number specified in the *Starting Run Number* field is `6`, the default value of *Number of Bins* is calculated as (100+6-1 = 105).

12. (Optional) Select the *Save Process Data* check box to save process parameter information in the ADE XL results database.

By default, this check box is selected.

Saving process parameter information in the ADE XL results database allows you to view sensitivity results. For more information, see Viewing Sensitivity Results on page 327.

Note: Saving process parameter information for designs with large number of process variables will increase the simulation run time.

13. (Optional) Select the *Save Mismatch Data* check box to save mismatch parameter information in the ADE XL results database so that it can be further used for viewing mismatch contribution data.

By default, this check box is deselected and ADE XL does not save mismatch parameters and their values and mismatch contribution analysis cannot be run.

Note: Saving mismatch parameter information for designs with large number of mismatch variables increases the simulation run time.

14. (Optional) If you have created a reference point and you want Monte Carlo points to be sampled around the reference point, select the *Use Reference Point* check box. For more information about reference points, see the *Virtuoso Analog Design Environment GXL User Guide*.

Note: If you select the *Use Reference Point* check box, ensure that the *# Point Sweeps* check box on the Run Summary pane is selected.

15. (Optional) Select the *Run Nominal Simulation* check box if you want the Spectre simulator to perform a nominal simulation run before starting the main Monte Carlo loop of iterations. If any errors are encountered during the nominal simulation run (for example, convergence problems, incorrect expressions, and so on), the Monte Carlo analysis is stopped. This allows you to stop unnecessary simulations from running if the nominal simulation fails.

Deselect the *Run Nominal Simulation* check box if you want Spectre to run the Monte Carlo iterations without performing a nominal simulation. If any errors are encountered during the Monte Carlo iterations, Spectre will issue a warning and continue with the next iteration of the Monte Carlo loop.

Note: If you are running a distributed Monte Carlo simulation, nominal simulation is performed only for the first job that is submitted by ADE XL.

16. (Optional) If you want to save raw data (`psf` files) for every Monte Carlo iteration so that you can perform postprocessing operations (like plotting, printing, annotation, and re-evaluation) on individual iterations, select the *Save Data to Allow Family Plots* check box.

If you do not select this check box:

- ❑ The program retains the `psf` data for the last iteration only.
- ❑ Signals and waveform expressions will not be plotted. Only histograms for scalar expressions will be plotted.

Note: The program calculates and saves scalar results for every run regardless of whether you select this check box or not.

Important

If you select this check box, the amount of raw data the program saves can be very large. You can reduce disk storage requirements, by saving only the nodes and terminals you reference in your output expressions. See “Saving All Voltages or Currents” on page 412 for information about saving nodes and terminals.

17. (Optional) If you want to specify a different seed for the Monte Carlo analysis, select the *Monte Carlo Seed* check box and type the seed number.

By always specifying the same seed, you can reproduce a previous experiment. If you do not specify a seed, the value `12345` is used.

18. (Optional) If you want to specify a starting run number, select the *Starting Run #* check box and type the starting run number.

The starting run number specifies the run that Monte Carlo begins with. By specifying this number, you can reproduce a particular run or sequence of runs from a previous experiment (for example, to examine an earlier case in more detail).

Note: To reproduce a run or sequence of runs, you need to specify the same value in the *Starting Run #* and the *Monte Carlo Seed* fields.

19. By default, mismatch variations are applied to all subcircuit instances in the design. Click the *Specify Instances/Devices* button to specify the sensitive instances and devices you want to either include or exclude for applying mismatch variations. For more information, see Including or Excluding Instances and Devices for Applying Mismatch Variations on page 276.

20. Click *OK*.

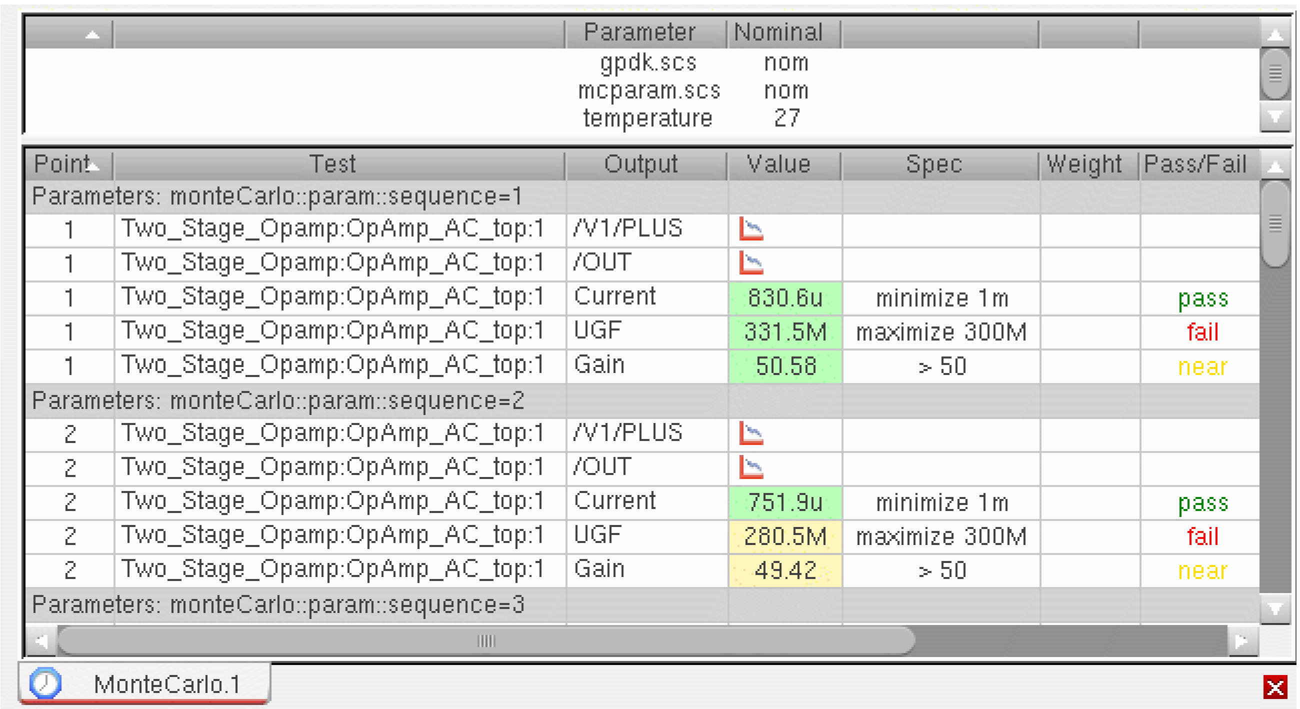
21. On the Run toolbar, click the *Run Simulation* button.

ADE XL calculates the number of simulations to be run. If the number exceeds the threshold limit specified using the warnWhenSimsExceed environment variable, a warning message is displayed to confirm if the run is to be continued.

After the run is complete, ADE XL displays the results in the *Yield* view on the Results tab of the Outputs pane. Results are reported for each test, corner and measurement expression and the total yield is reported for the circuit.

You can also use the Summary view or the Detail view on the Results tab to view the detailed results for the Monte Carlo analysis. In the Summary view and Detail view, the results for each Monte Carlo sample is displayed under a row named
`Parameters: monteCarlo::param::sequence=`*n*

where *n* is the number of the sample. For example, in the following figure of the Detail view, the results under the row `Parameters: monteCarlo::param::sequence=`*1* displays the results for the first sample.



To view the description of the each column in the Details view of the Results tab, refer to Hiding and Showing Data on the Results Tab. Note that in some cases, the results are shown as `Canceled`. To know more about this status, refer to Starting a Simulation.

Note: The program converts waveforms that have `mcparamset` as the sweep name to histograms and plots each such waveform in a separate subwindow.

The program also displays a pass/fail type of histogram plot for the results. For more information about plotting histograms for Monte Carlo results, see Plotting Histograms on page 316.

Debugging in Debug Environment

If any particular data point fails or shows undesired results, you can select the result in the Detail view and load it in the debug environment. Here, you can modify the setup of that data point to debug it and run a single Monte Carlo simulation to verify the results, and bring back the updated test to the active adexl view.

See also:

- Including or Excluding Instances and Devices for Applying Mismatch Variations on page 276
- Stopping Monte Carlo Based on the Target Yield on page 293
- Running Incremental Monte Carlo Analyses on page 297
- Generating Plots, Tables, and Reports on page 310
- Viewing Sensitivity Results on page 327
- Viewing Statistical Parameters for Monte Carlo Samples on page 327
- Viewing Monte Carlo Results on page 297

Including or Excluding Instances and Devices for Applying Mismatch Variations

By default, mismatch variations are applied to all subcircuit instances in the design. You can use the Specify Instances for Mismatch form to do one of the following:

- Select specific instances and devices to which mismatch variations must be applied. Mismatch variations will also be applied to all subcircuits instantiated under the selected instances.
- Select specific instances and devices to which mismatch variations must not be applied. Mismatch variations will also not be applied to all subcircuits instantiated under the selected instances.

Important

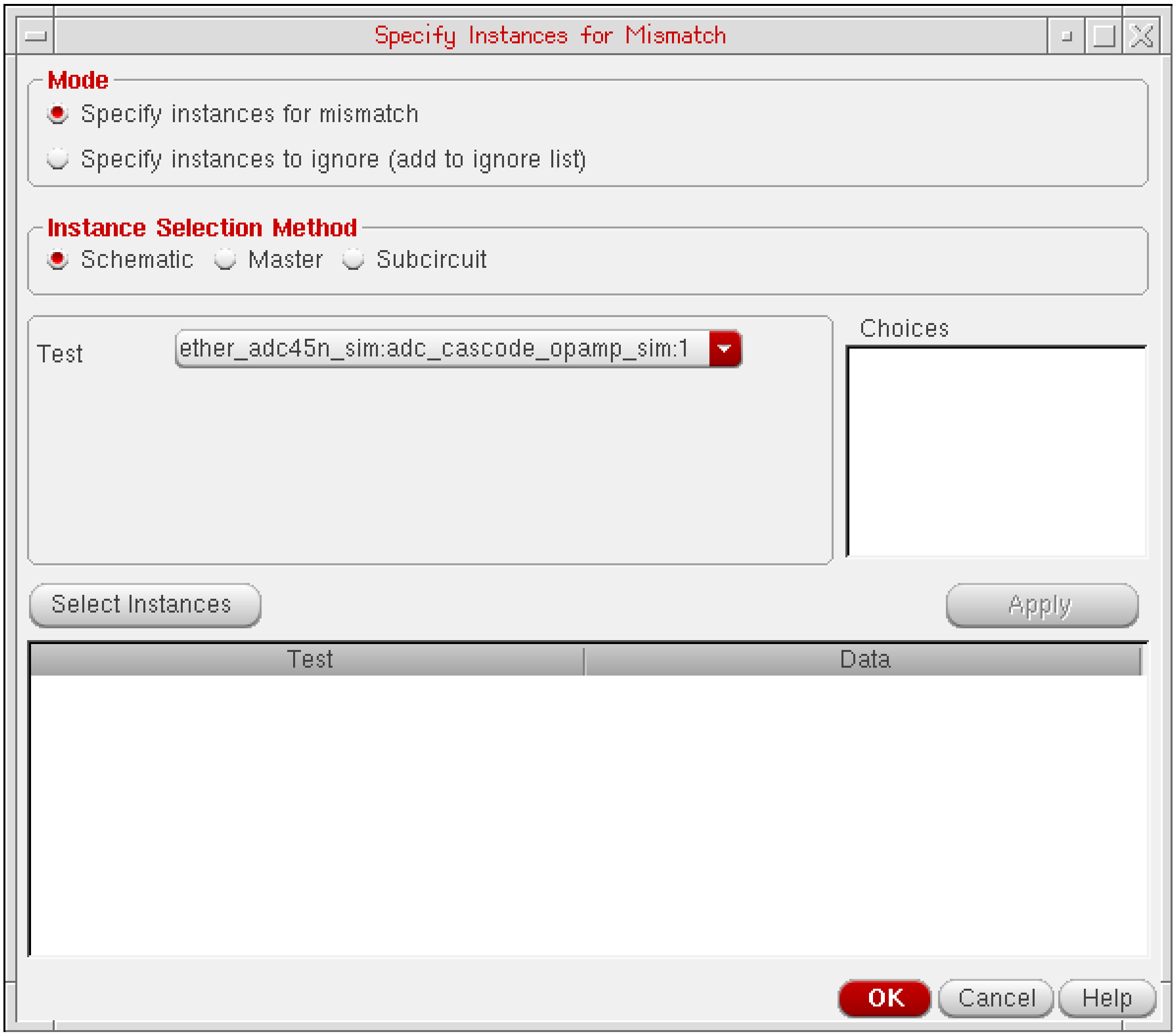
If multiple ADE XL tests are enabled, ensure that the same single top level DUT (Device Under Test) instance is selected for all the tests, so that you get the same statistical variation across all tests. This allows you to accurately predict the total yield across multiple tests.

To include or exclude specific instances and devices for applying mismatch variations, do the following:

1. Click the *Specify Instances/Devices* button on the Monte Carlo form.

   Note: If the button name appears as *Specify Instances/Devices (Not Specified)*, it indicates that no instances or devices are selected. Therefore, mismatch variations are applied to all subcircuit instances in the design. If the button name appears as *Specify Instances/Devices (Exists)*, it indicates that you have selected specific instances and devices to include or exclude for applying mismatch variations.

The Specify Instances for Mismatch form appears.



2. Do one of the following:

Select	**To**
Specify instances for mismatch	Select the instances and devices to which mismatch variations must be applied. In this case, mismatch results will be displayed only for the selected instances and devices.

Select	To
Specify instances to ignore (add to ignore list)	Select the instances and devices to which mismatch variations must not be applied. In this case, mismatch results will not be displayed for the selected instances and devices. The mismatch results will be displayed for all other instances and devices.

3. Do one of the following:

Select	To
Schematic	Select the instances you want to include or exclude for applying mismatch variations in the schematic. For more information, see Selecting Schematic Instances for Applying Mismatch Variations on page 279.
Master	Select instances of cellviews you want to include or exclude for applying mismatch variations. All the instances within the selected cellviews will be included or excluded. For more information, see Selecting Instances of Cellviews for Applying Mismatch Variations on page 283.
Subcircuit	Select the subcircuit instances you want to include or exclude for applying mismatch variations. All the instances within the selected subcircuits will be included or excluded. For more information, see Selecting Subcircuit Instances for Applying Mismatch Variations on page 288.

Important

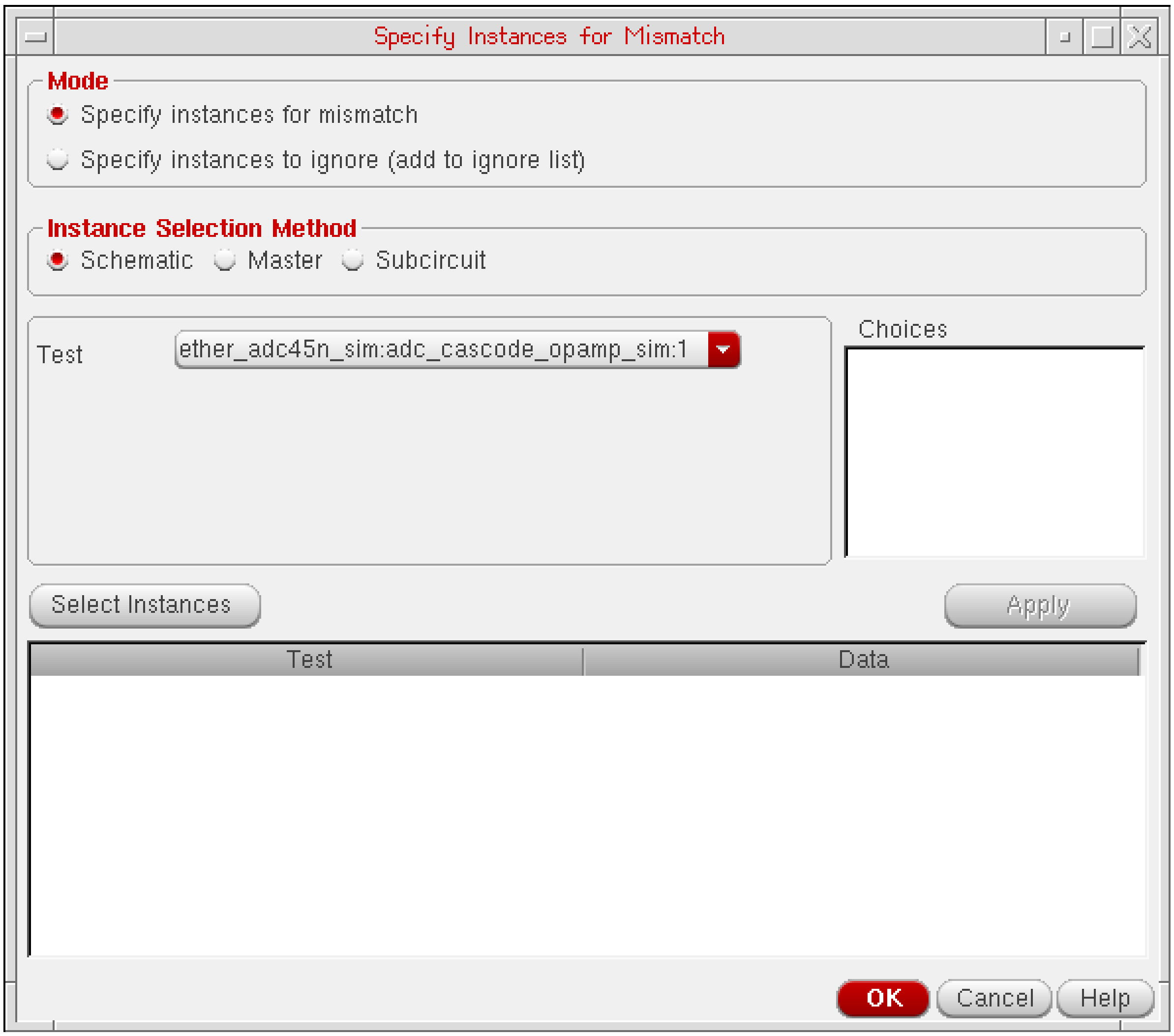
You can either select the instances and devices to be included or excluded for applying mismatch variations. You cannot select instances and devices to be included and excluded at the same time.

4. Click *OK* to save the changes and return to the Monte Carlo options form.

Selecting Schematic Instances for Applying Mismatch Variations

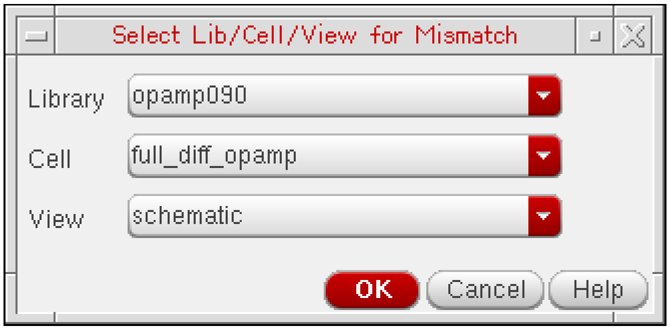
To select instances you want to include or exclude for applying mismatch variations in the schematic, do the following:

1. Select the *Schematic* option.



2. In the *Test* drop-down list, choose the test for which you want to select instances.

To select instances for all the tests that are enabled in the Data View pane, choose *All Tests* in the *Test* drop-down list. If you choose *All Tests*, the Select Lib/Cell/View for Mismatch form appears.



Do the following to open the schematic view from which you want to select instances for all the tests that are enabled in the Data View pane.

a. Use the *Library*, *Cell* and *View* drop-down lists to select the schematic view.

Note: The *Library*, *Cell* and *View* drop-down lists display only the cellviews that are used in the designs for all the tests.

b. Click *OK*.

The selected schematic view is opened in a new tab.

3. If a test name is selected in the *Test* drop-down list, click the *Select Instances* button. The schematic for the test is opened in a new tab.

4. In the schematic, select one or more instances.

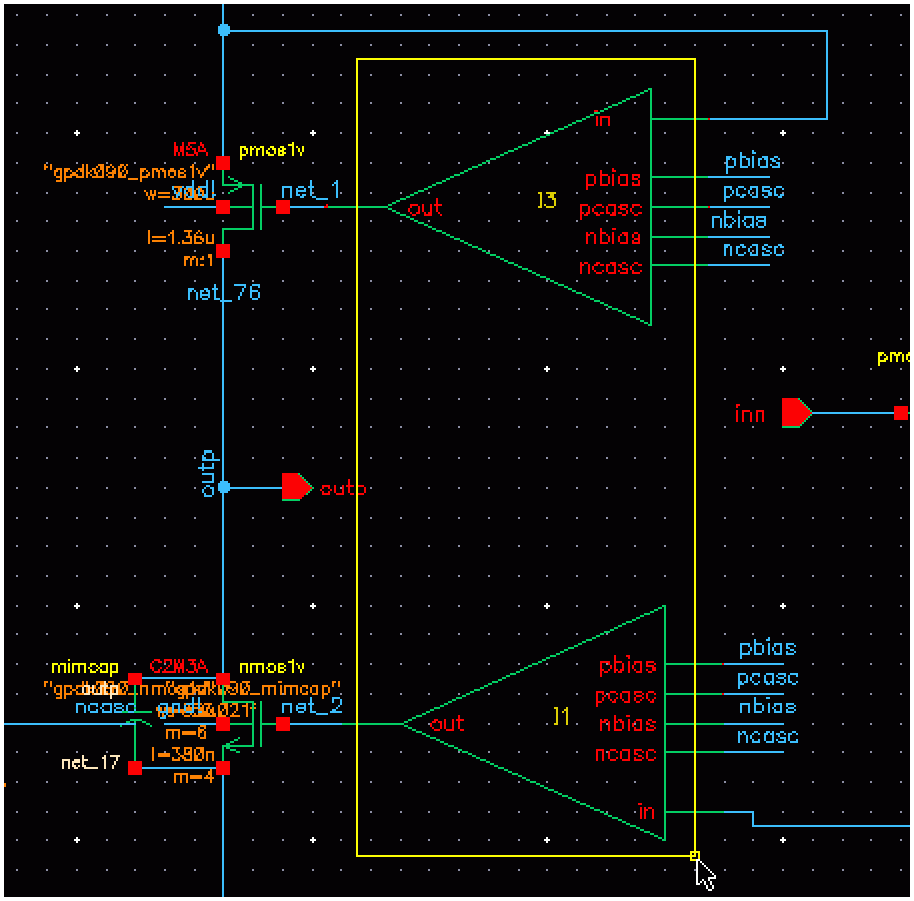
To select more than one instance at a time, do one of the following:

❑ Hold down the *Shift* key and click on instances.

❑ Click and drag the mouse over the instances you want to select.

All the instances that are within the yellow bounding box that appears are included in the selection.

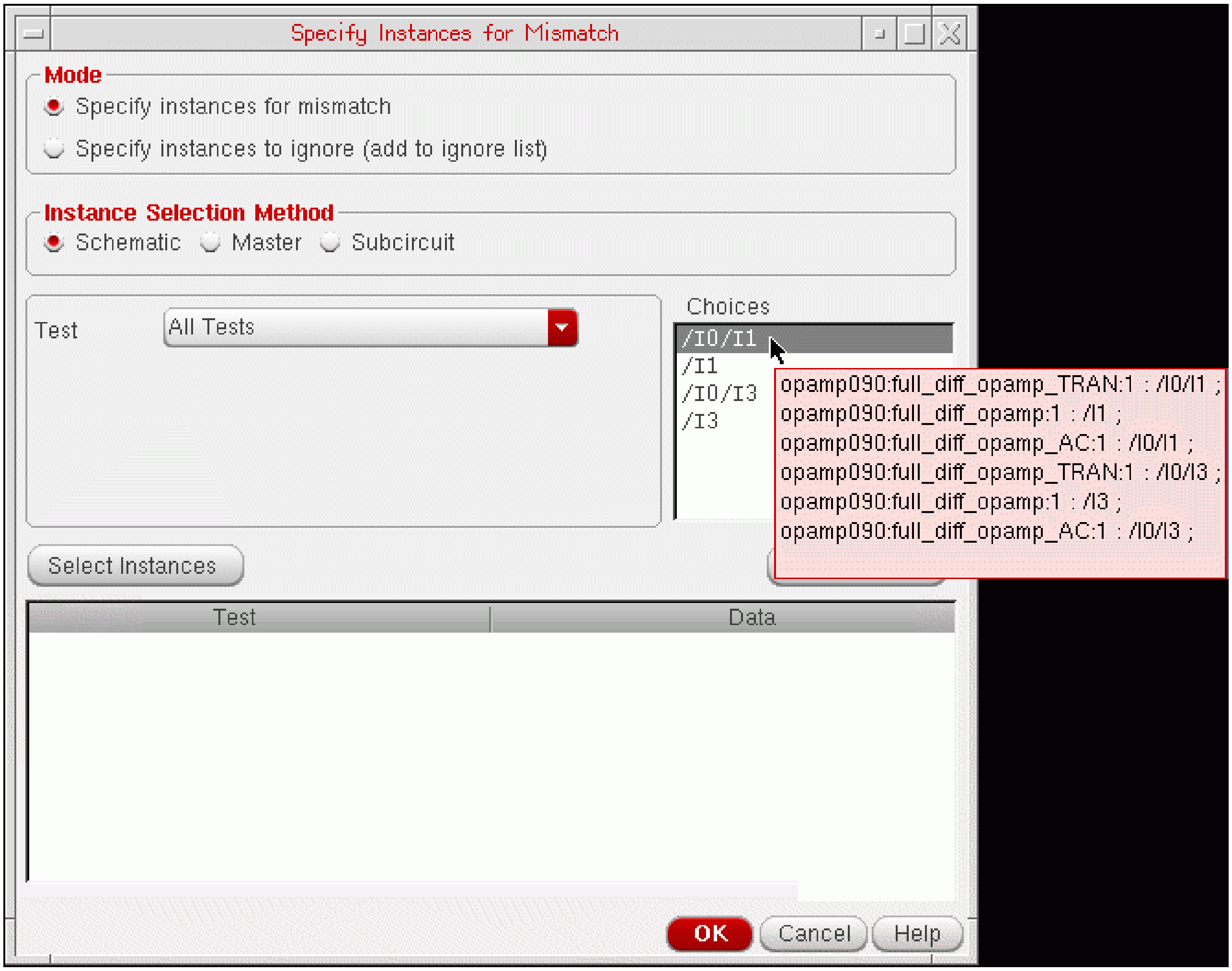
In the following example, instances `I1` and `I3` that are within the yellow bounding box are included in the selection.



5. Press the *Esc* key when you are done.

   The selected instances are displayed in the *Choices* field.

   If *All Tests* is selected in the *Test* drop-down list, the instance names of the selected instances in the designs for all the tests are displayed in the *Choices* field. You can place the mouse pointer in the *Choices* field to view the instance names of the selected instances in the design for each test.

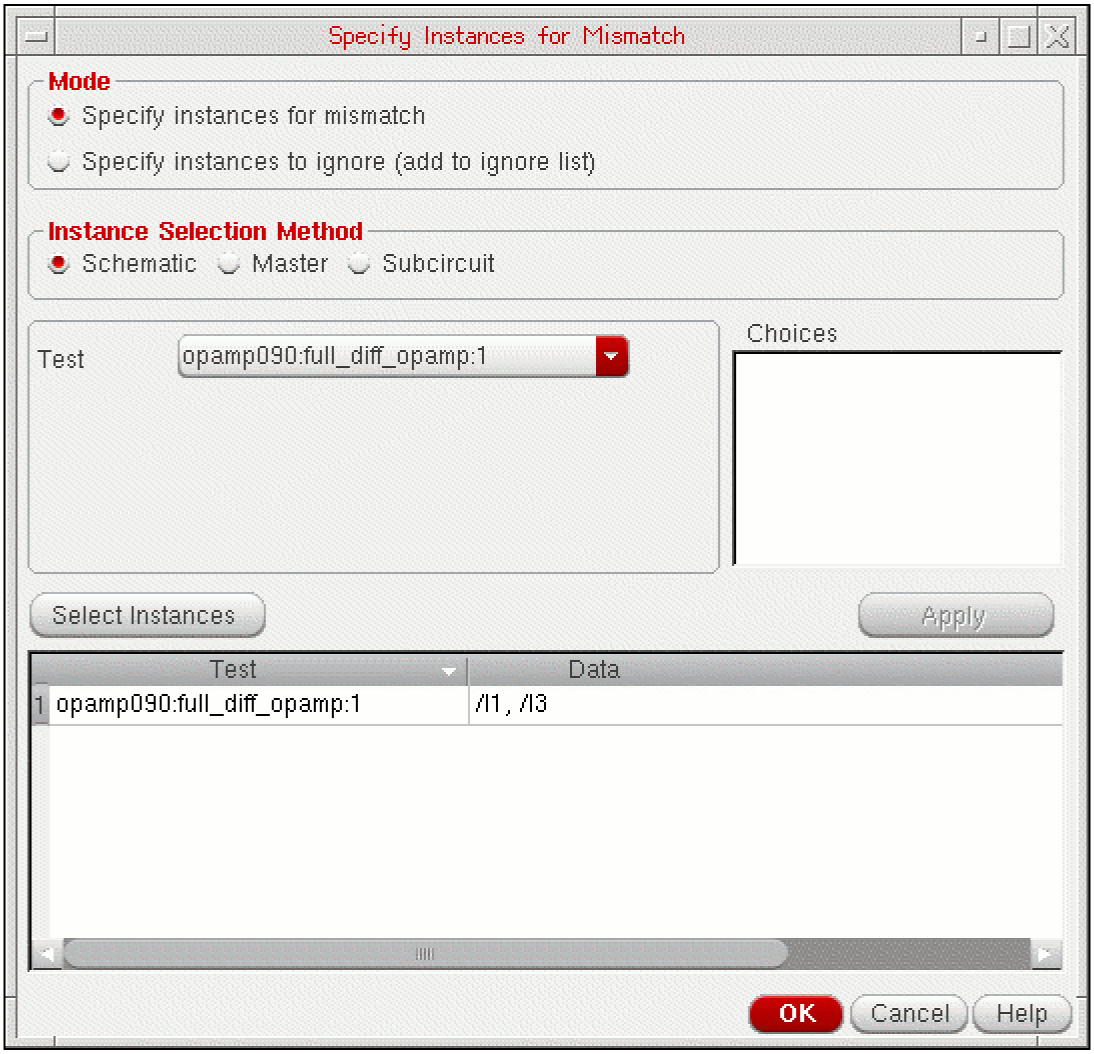


6. In the *Choices* field, select the instances you want to include or exclude for applying mismatch variations.

   To select multiple instances, hold down the *Shift* key (for contiguous selection) or the *Ctrl* key (for noncontiguous selection) and click the next instance to add it to the selection set.

7. Click *Apply*.

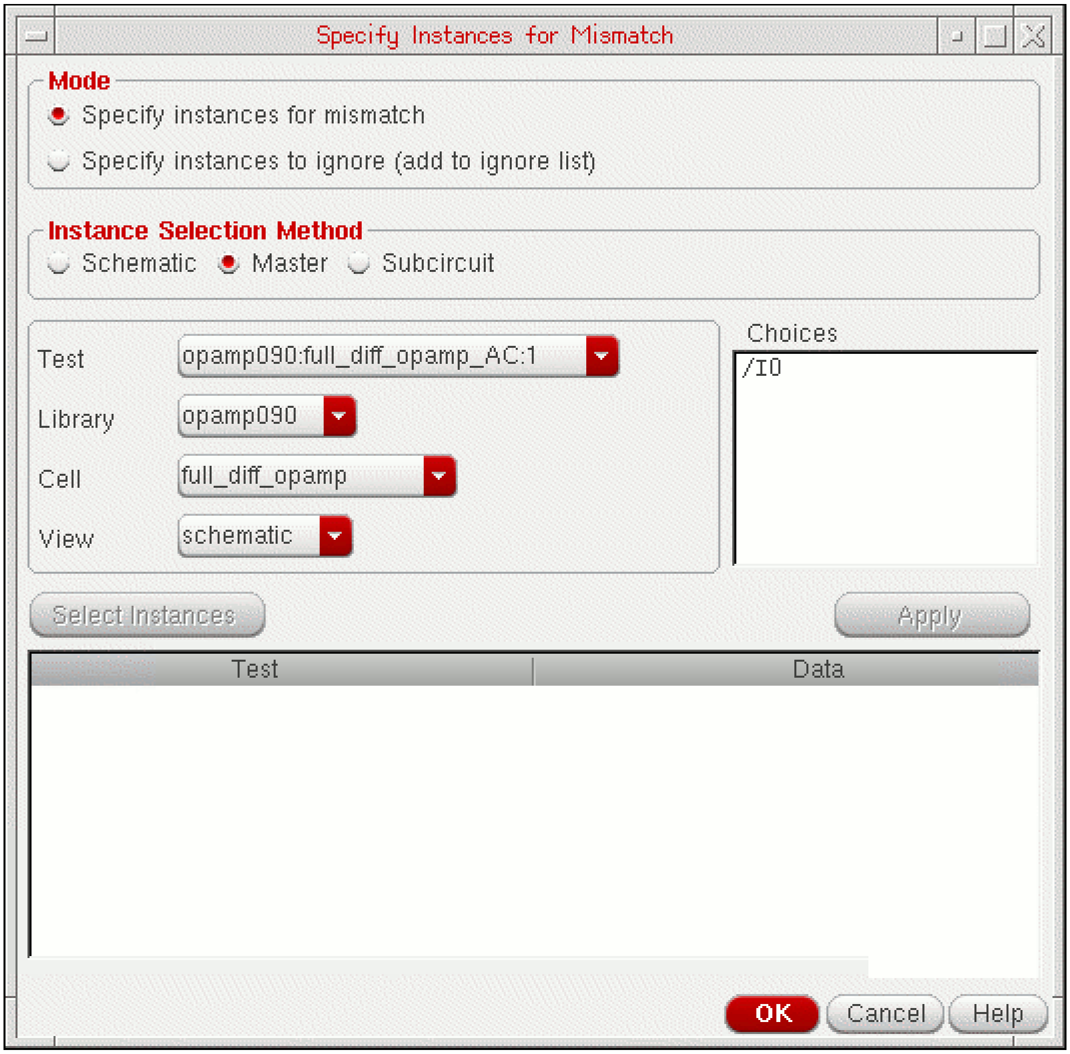
The selected instances are added for the test. For example, in the following figure, the instances `I1` and `I3` are added for the `opamp090:full_diff_opamp:1` test.



Selecting Instances of Cellviews for Applying Mismatch Variations

To select instances of cellviews you want to include or exclude for applying mismatch variations, do the following:

1. Select the *Master* option.



2. In the *Test* drop-down list, choose the test for which you want to select instances of cellviews.

   To select instances for all the tests that are enabled in the Data View pane, choose *All Tests* in the *Test* drop-down list.

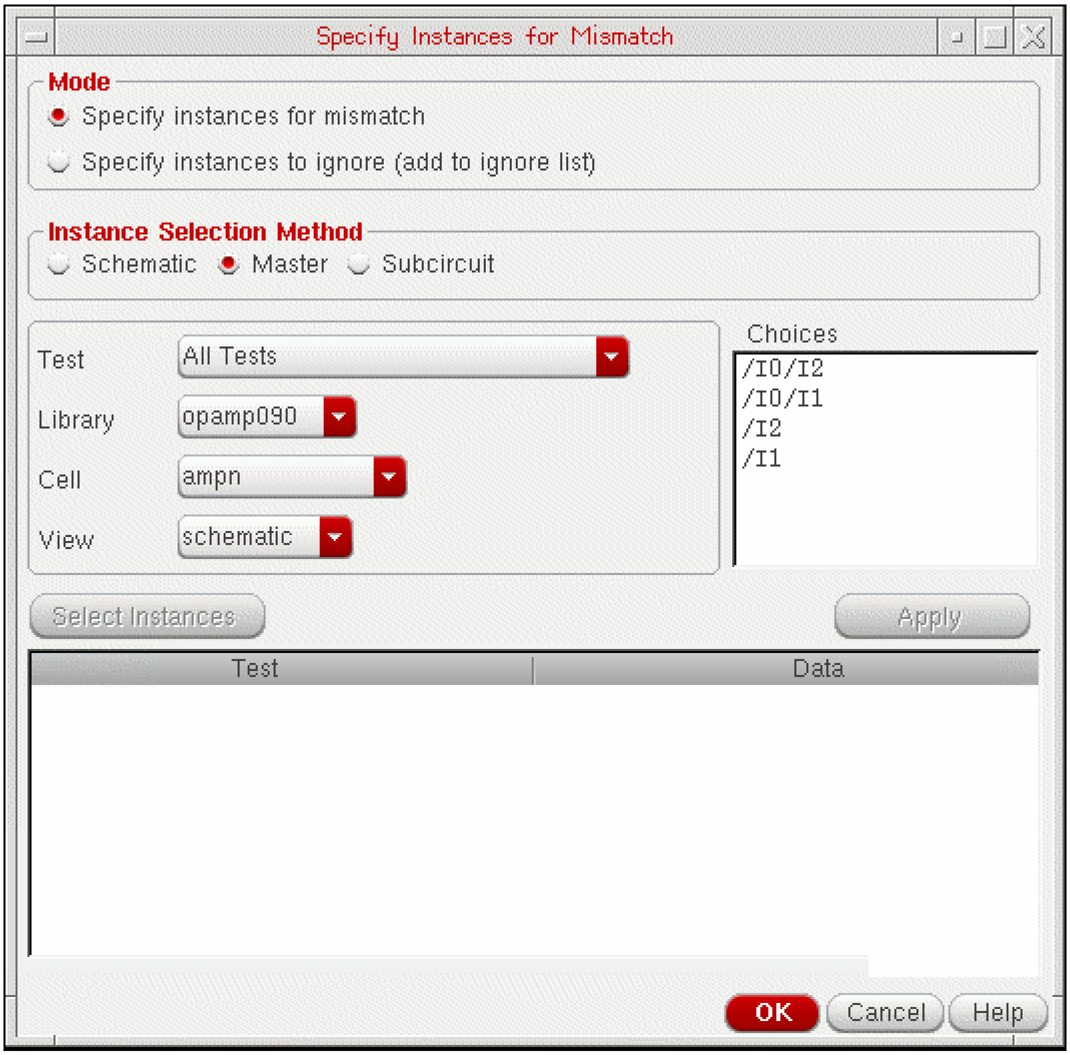
3. Use the *Library*, *Cell* and *View* drop-down lists to select the library, cell and view in which the cellview exists.

   All the instances of the cellview in the design for the test are displayed in the *Choices* field.

   Note the following:

- You can set the `ignoredLibsForDUT` environment variable in your `.cdsenv` file to specify the libraries that should not be displayed in the *Library* drop-down list.
- By default, the following views are not displayed in the *View* drop-down list:
  - adestate
  - adexl
  - config
  - physConfig
  - schematicSymbol
- If *All Tests* is selected in the *Tests* drop-down list, the *Choices* field displays the instance names of the cellview in the designs for all the tests.

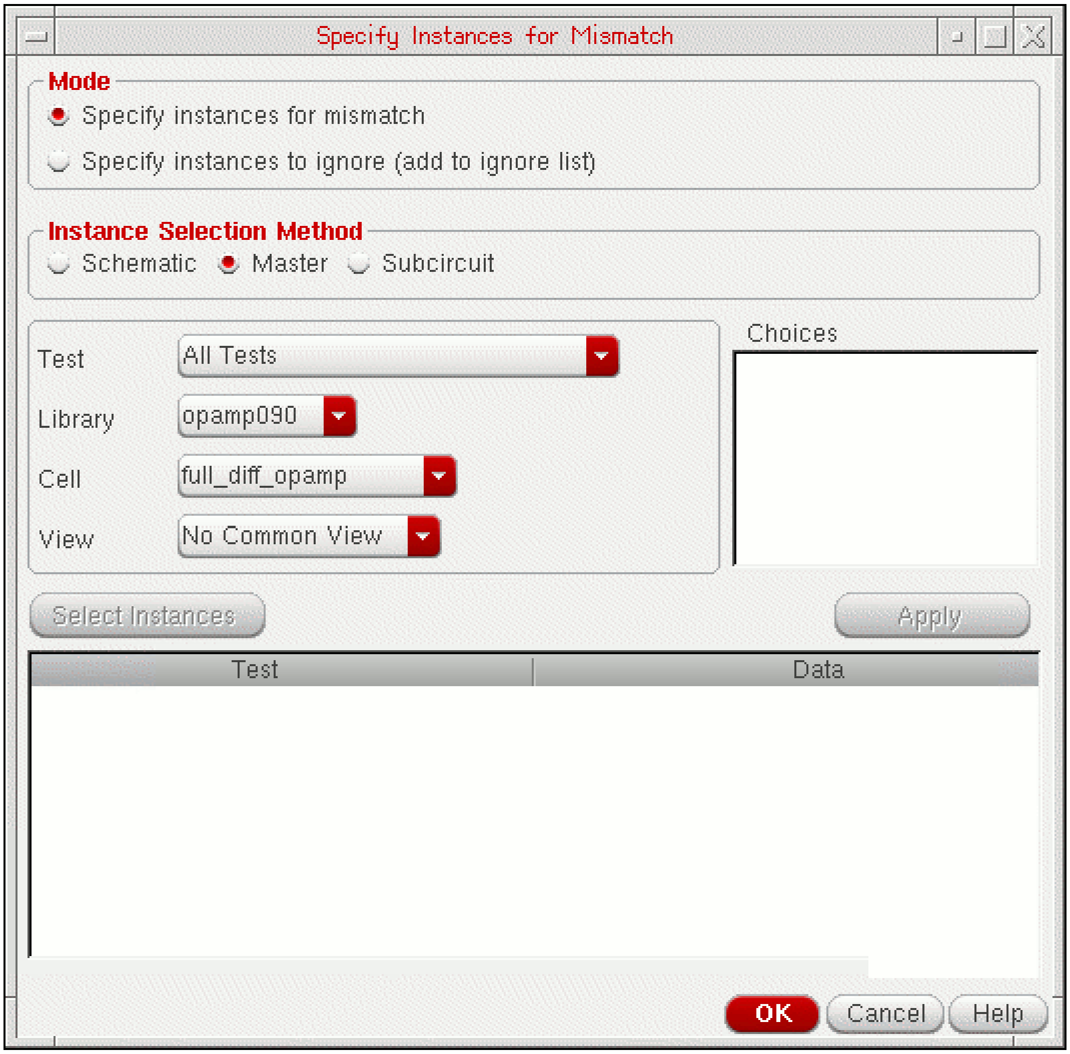
For example, in the following figure, the *Choices* field displays the instance names of the `schematic` cellview of the `ampn` cellview in the designs for all the tests.



You can place the mouse pointer on the *Choices* field to view the instance name of the cellview in the design for each test.

❑ If *All Tests* is selected in the *Tests* drop-down list, you can add only instances of cellviews that exist in the designs for all the tests. If instances of a cellview do not exist in the designs for all the tests, the *View* field displays the text *No Common View* indicating that you cannot select instances of the cellview.

For example, in the following figure, the *View* field displays the text *No Common View* because none of the cellviews of the `full_diff_opamp` cell in the `opamp090` library are instantiated in the designs for all the tests.

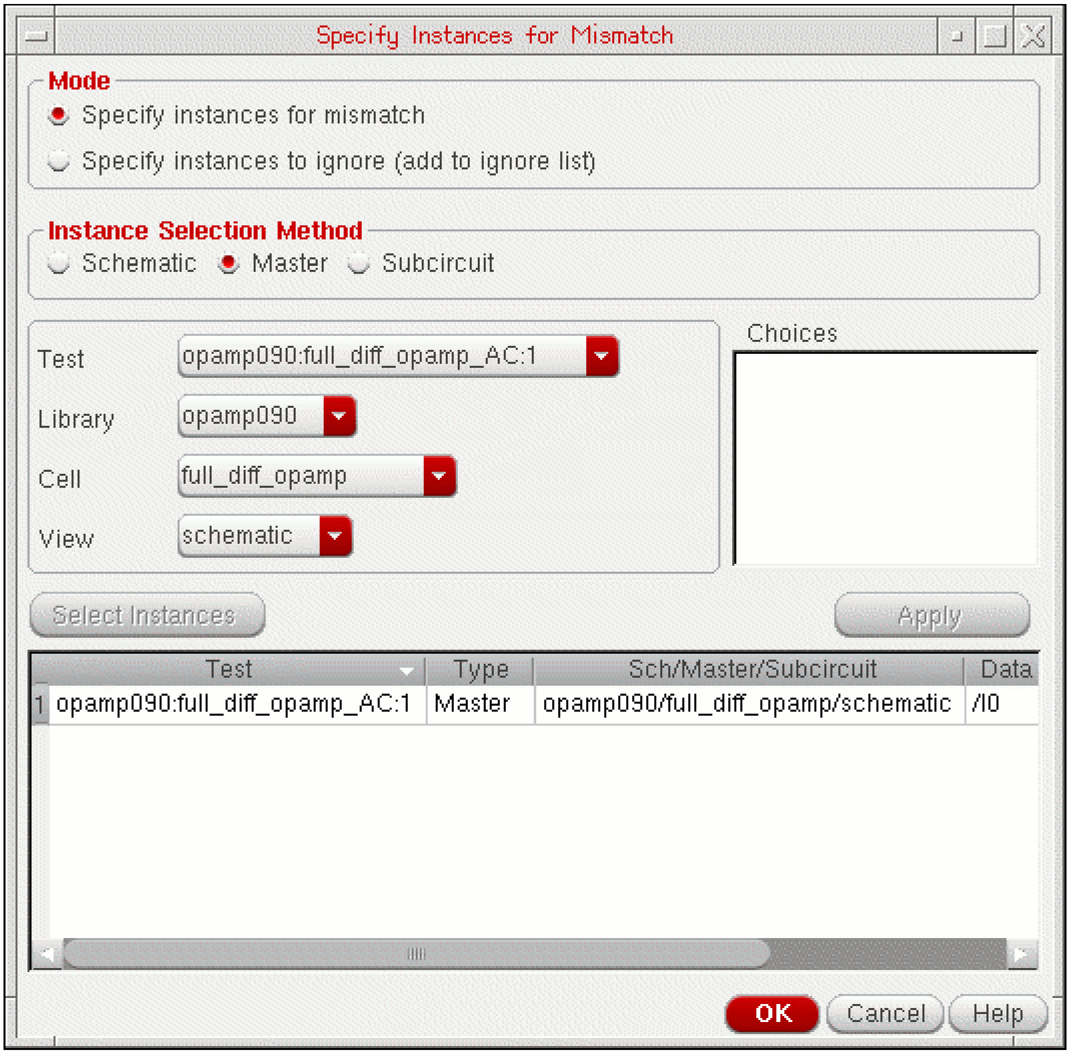


4. In the *Choices* field, select the cellview instances you want to include or exclude for applying mismatch variations.

   To select multiple instances, hold down the *Shift* key (for contiguous selection) or the *Ctrl* key (for noncontiguous selection) and click the next instance to add it to the selection set.

5. Click *Apply*.

The selected cellview instances are added for the test. For example, in the following figure, the cellview instance `/I0` is added for the `opamp090:full_diff_opamp_AC:1` test.

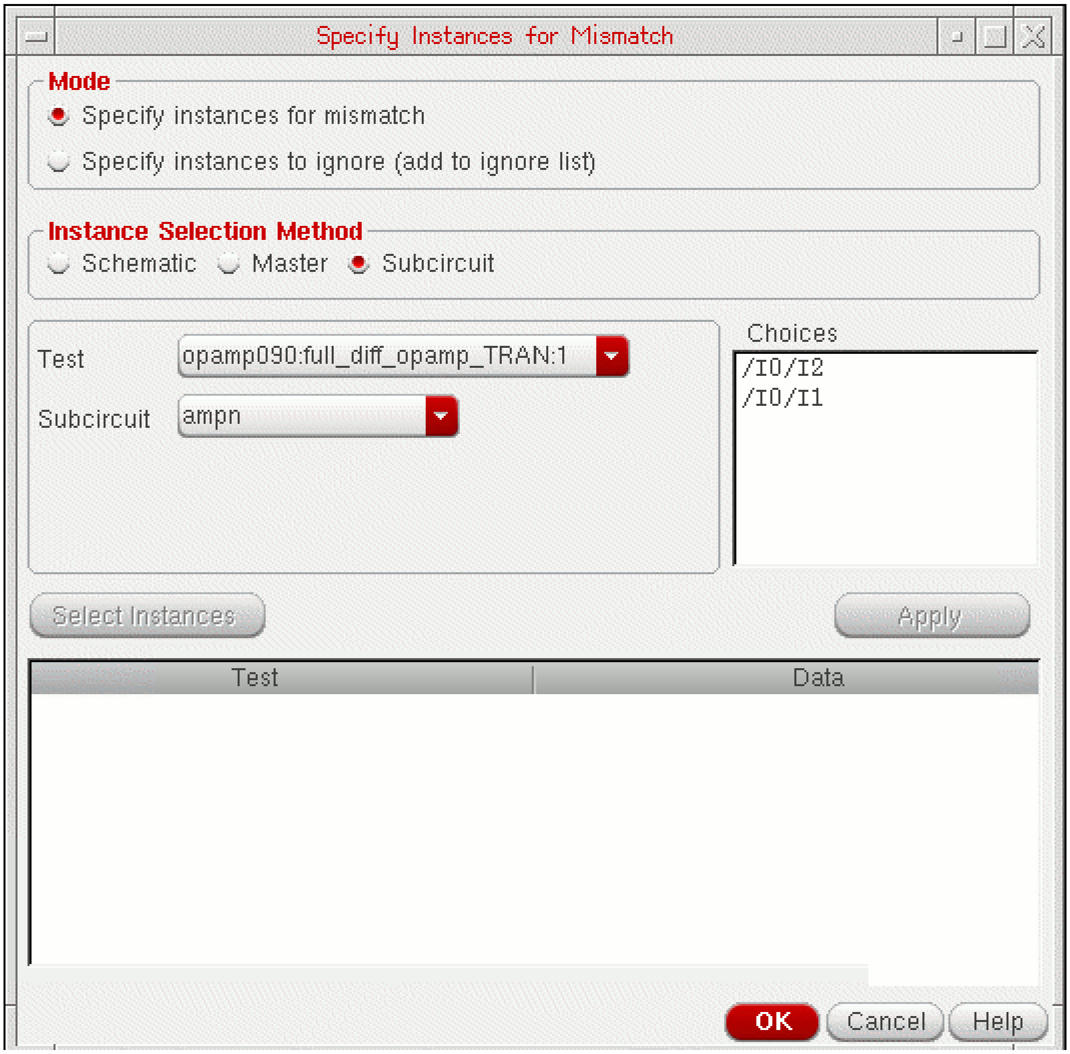


Note: By default, the *Type* and *Sch/Master/Subcircuit* columns are not visible. To view these columns, right-click on the title bar and select the name of the column to be displayed in this table.

Selecting Subcircuit Instances for Applying Mismatch Variations

To select the subcircuit instances you want to include or exclude for applying mismatch variations, do the following:

1. Select the *Subcircuit* option.



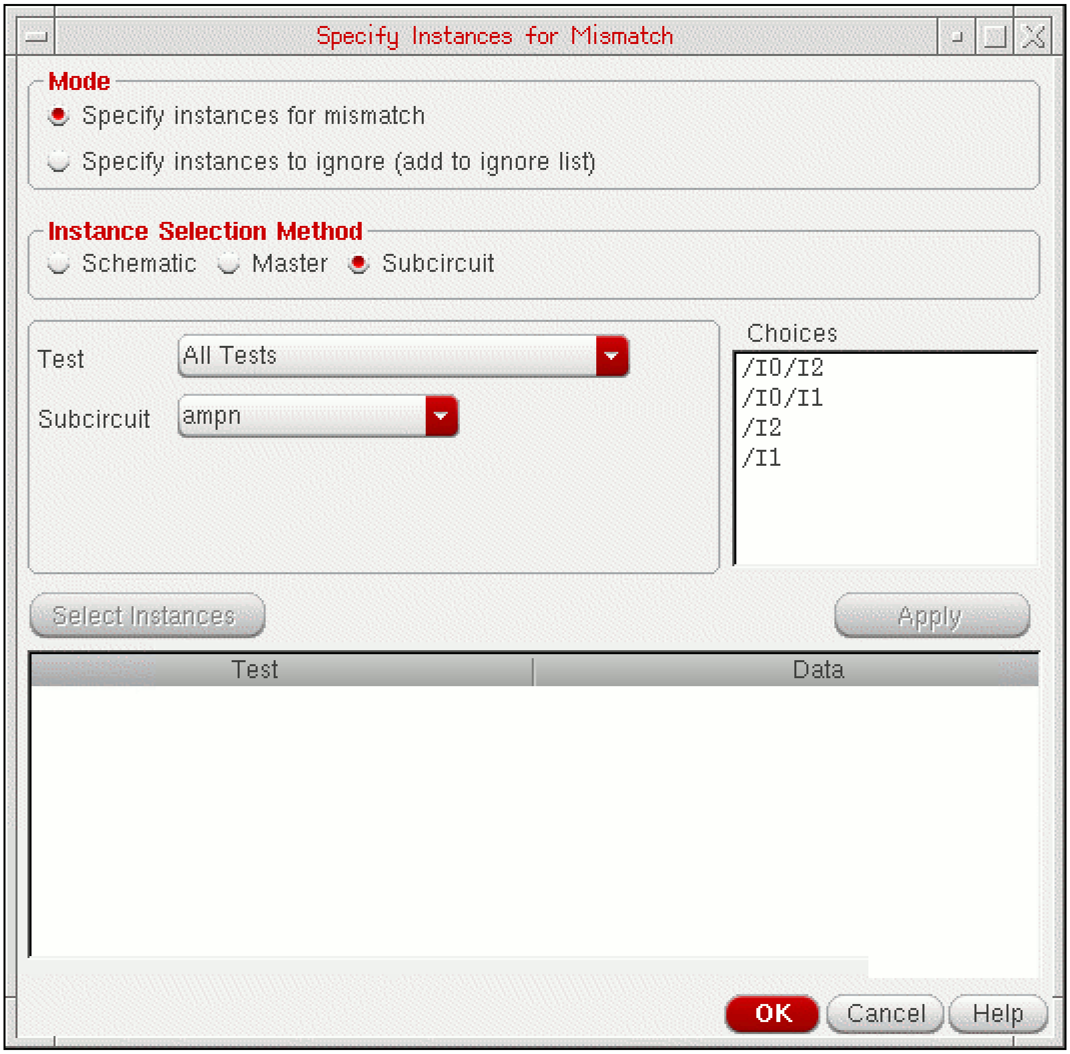
2. In the *Test* drop-down list, choose the test for which you want to select subcircuit instances.

   To select subcircuit instances for all the tests that are enabled in the Data View pane, choose *All Tests* in the *Test* drop-down list.

3. In the *Subcircuit* drop-down list, choose the subcircuit whose instances you want to select.

   All the instances of the subcircuit in the design are displayed in the *Choices* field.

   Note: If *All Tests* is selected in the *Tests* drop-down list, the *Choices* field displays the instance names of the subcircuit in the designs for all the tests. For example, in the following figure, the *Choices* field displays the instance names of the `ampn` subcircuit in the designs for all the tests.



Tip

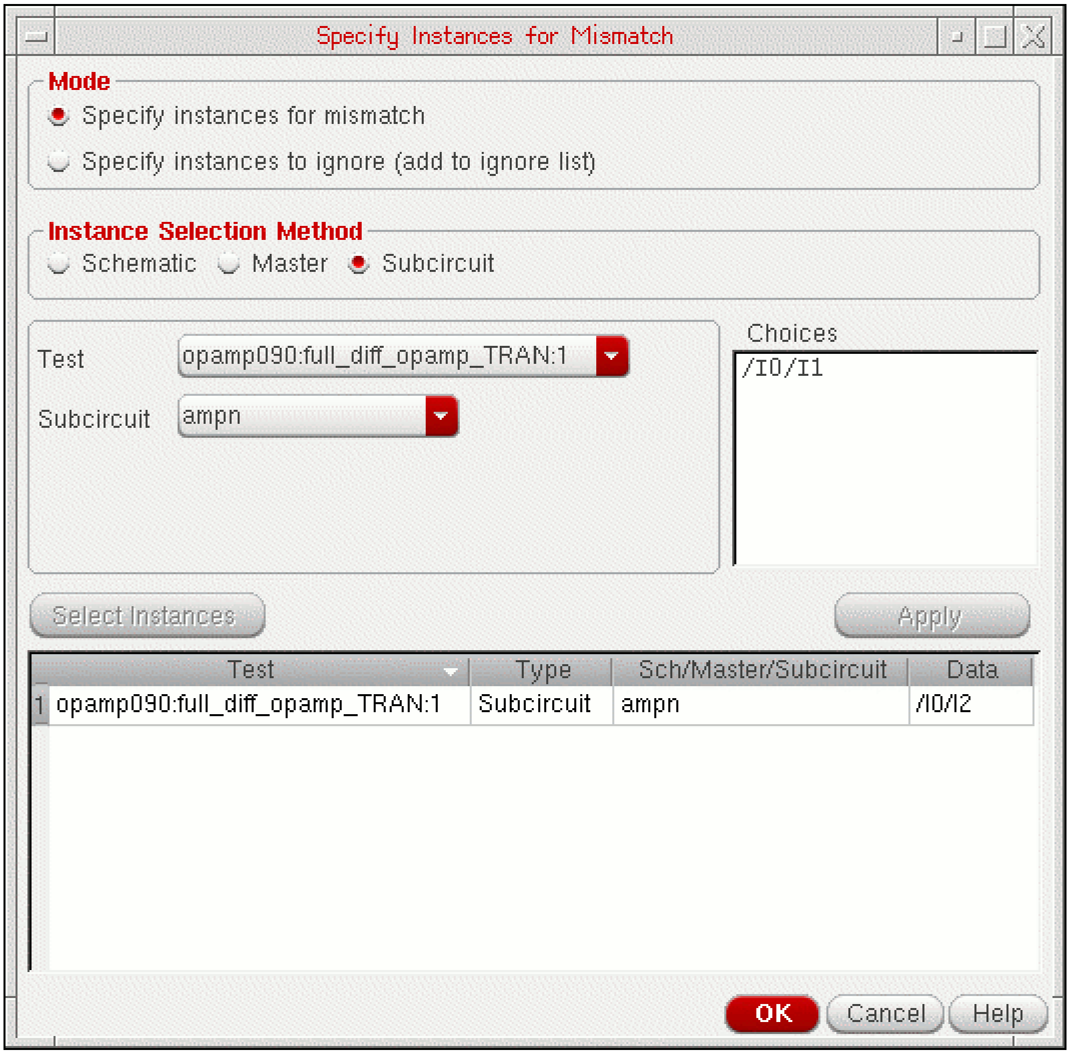
You can place the mouse pointer in the *Choices* field to view the instance name of the subcircuit in the design for each test.

4. In the *Choices* field, select the subcircuit instances you want to include or exclude for applying mismatch variations.

   To select multiple instances, hold down the *Shift* key (for contiguous selection) or the *Ctrl* key (for noncontiguous selection) and click the next instance to add it to the selection set.

5. Click *Apply*.

The selected instances are added for the test. For example, in the following figure, the instance `/I0/I2` is added for the `opamp090:full_diff_opamp_TRAN:1` test.



Note: By default, the *Type* and *Sch/Master/Subcircuit* columns are not visible. To view these columns, right-click on the title bar and select the name of the column to be displayed in this table.

Deleting Instances Selected for Applying Mismatch Variations

To delete an instance, do the following:

➡ Right-click on the row for the instance and choose *Delete*.

To delete multiple instances, hold down the *Shift* key (for contiguous selection) or the *Ctrl* key (for noncontiguous selection), click the rows for the instances you want to delete, then right-click and choose *Delete*.

To delete all instances, do the following:

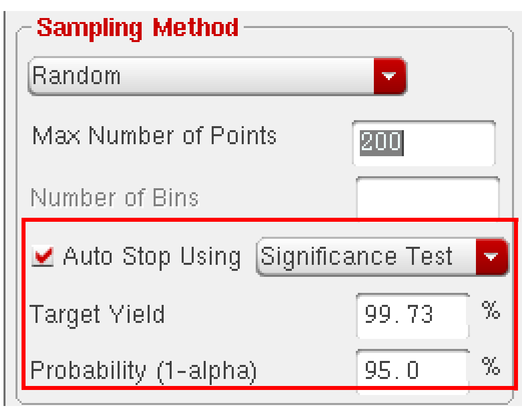
1. Right-click on the row for an instance and choose *Select All*.
2. Right-click on the row for an instance and choose *Delete*.

Stopping Monte Carlo Based on the Target Yield

The accuracy of the yield estimate from the Monte Carlo run increases by simulating a large number of samples. However, running a large number of simulations can consume a lot of time and resources if a design is large. Auto Stop Using Significance Test is a method to automatically identify the appropriate number of samples for Monte Carlo analysis to simulate based on a target yield estimate. The Monte Carlo run will stop automatically without running additional simulations as soon as it can be determined that the yield is above or below the target.

To enable the significance test stopping criteria, select `Significance Test` from the *Auto Stop Using* drop-down list and specify the following values:

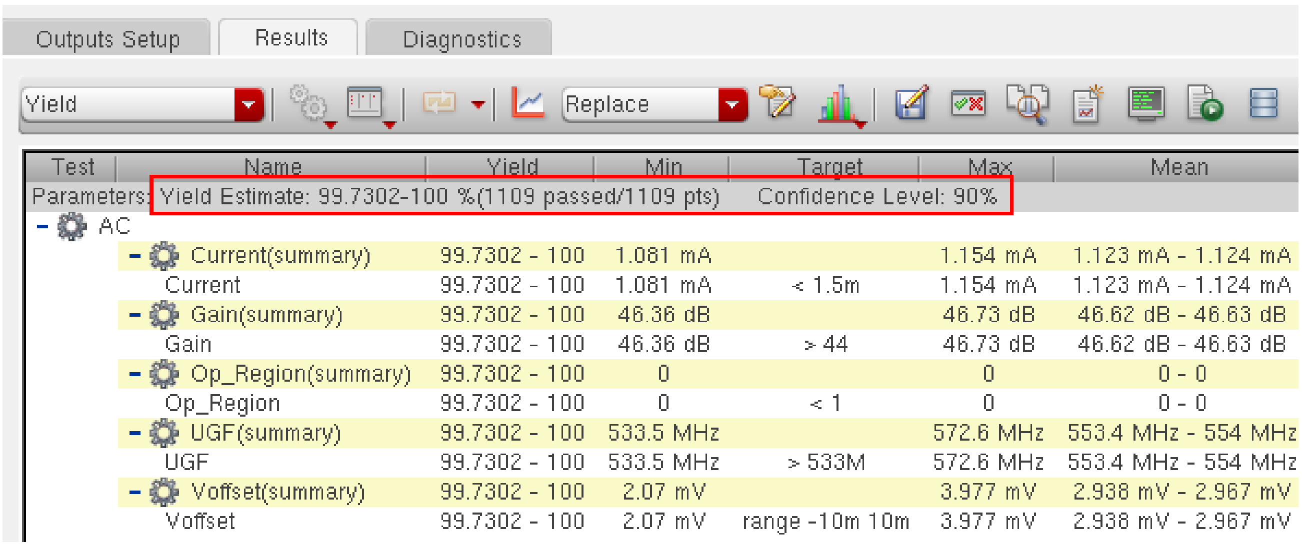
- *Target Yield*: This is the target yield value that you want to achieve for your design. Default value for this field is 99.73%.
- *Probability (1-alpha)*: This defines the significance level. Specify a value between 50% and 100% (not including 100%). Probability values closer to 100% will require more simulations before the yield estimate can be determined to be lower or higher than the target. Smaller probability values will result in a looser test and require less simulations before auto stop is triggered. Default value = 95%.



If the significance test is enabled but the stopping conditions are never met, the analysis will stop after simulating the maximum number of samples specified in the Max Number of Points field. Auto Stop Using Significance Test is available for the Random and Low-Discrepancy Sequence Sampling methods.

The following results are reported when Auto Stop Using Significance Test is enabled:

- The *Yield* view on the Results tab displays the overall yield estimate and the confidence level in the gray row on top of the detailed results.

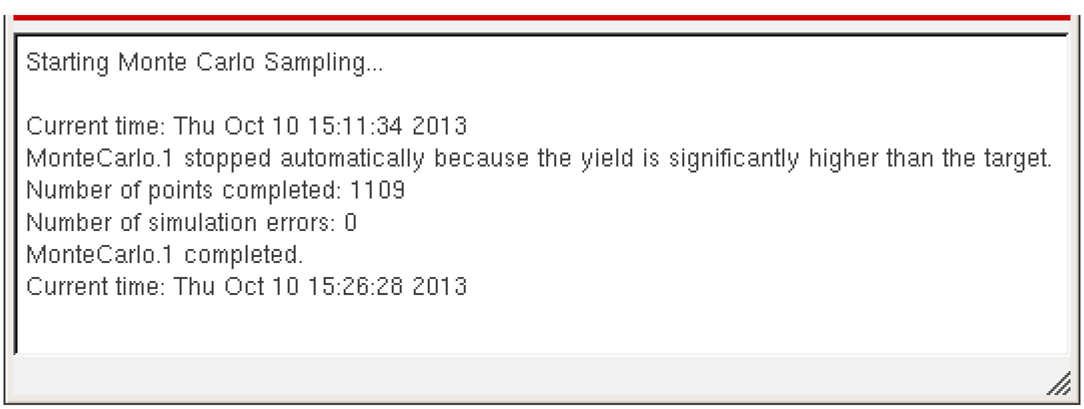


The Clopper-Pearson method is used to calculate the binomial confidence interval of the yield estimate. Monte Carlo will stop if either the upper bound of the yield estimate is lower than the yield target, or if the lower bound of the yield estimate is higher than the yield target. Since auto stop applies two one-sided tests relating the significance test to the confidence interval is as follows:

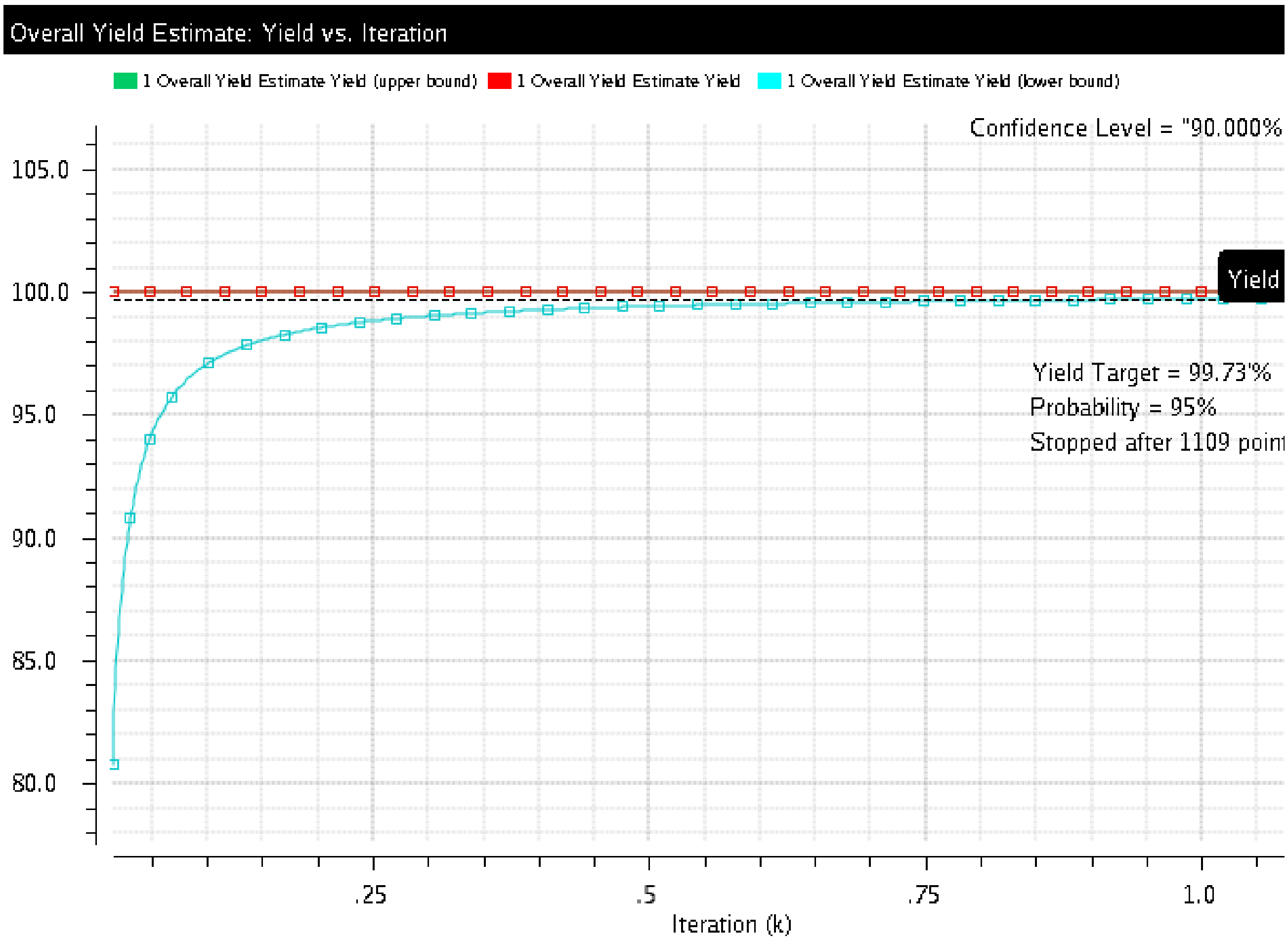
```
Confidence Level = 100 * (1 - 2*alpha)
```

If Auto Stop is not enabled, you can still analyze the results by specifying a confidence level. For more details, refer to <u>Viewing Data for a Specific Confidence Interval</u>.

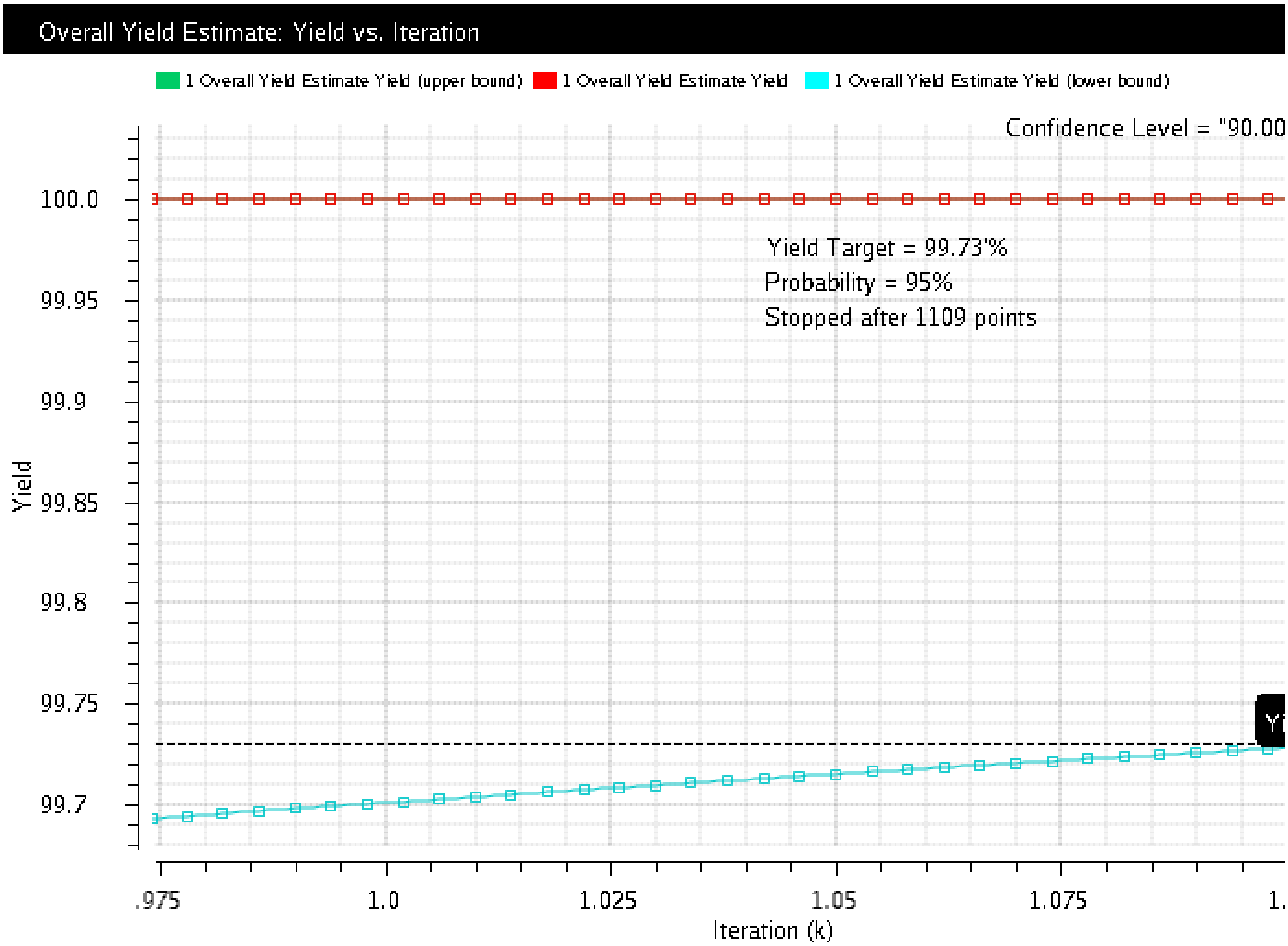
- The run log shows the run status and the number of samples for which the analysis was run. The figure given below shows a sample.



- The overall yield estimate plot shows a graph with the yield estimate and confidence interval at every iteration. To view the yield estimate plot, click on the Results tab, select the test name and choose Plot/Print Versus Iteration. Keep the default options and click *Apply* to plot a graph. The following figure shows a sample.



- In this example, there were no failed samples. Monte Carlo stopped after 1109 points. At this point, the lower bound of the yield estimate confidence interval is greater than the yield target.



Note: In a normal Monte Carlo run, if a single ICRP job is used, it may simulate a specific corner for all sample values before fully evaluating any sample over all corners. However, when the significance test is enabled, by default, an ICRP job keeps a maximum of 40 sample points outstanding (or in partially evaluated state). That is, if there is only one ICRP job, first it runs and completely evaluates first 40 samples for all corners and uses the success status for the significance test. After that, the job moves to the next 40 samples and uses them to run simulations for all corners, and so on. This helps in getting the results of the significance test earlier, before running all samples runs for all corners.

If required, you can change the maximum of sample points that can be outstanding for a corner at a given time by setting the maxOutstandingPoints environment variable.

Running Incremental Monte Carlo Analyses

You can use the Manual Tuning mode to run Monte Carlo incrementally. In this run mode, you can append the results of multiple Monte Carlo runs to the same history and results view.

Important Points to Note

Consider the following points while running incremental Monte Carlo analyses in the Manual Tuning run mode:

- In order to run incrementally, all Monte Carlo runs must be performed within the Manual Tuning mode. You cannot append to a run performed outside of the Manual Tuning mode.
- If the chosen sampling method is `Random`, for every subsequent Monte Carlo run, specify an incremented value in the *Starting Run Number* field. For example, if you run the first Monte Carlo run for 20 points, in the subsequent run, set the *Starting Run Number* field on the Monte Carlo form to `21`. Similarly, the value in the *Number of Points* field can vary across different Monte Carlo runs. For example, if you run the first Monte Carlo run for 20 points, for the second one, you can set the *Number of Points* field to 40.
- If the chosen sampling method is `Latin Hypercube`, specify a different seed value fin the *Monte Carlo Seed* field or each subsequent Monte Carol run.

For more details on the Manual Tuning run mode, refer to Manual Tuning in the *Analog Design Environment GXL User Guide*.

Viewing Monte Carlo Results

The default results view for the Monte Carlo Sampling run mode is the *Yield* view. This view displays the overall yield estimate based on the pass or fail status of all the specifications. For details on the different columns in this view, refer to the *Yield* view.

Also see:

- Managing Monte Carlo Results in the Yield View
- Plot/Print versus Iteration
- Plotting Histograms
- Plotting Scatter Plots

Managing Monte Carlo Results in the Yield View

The following topics explain how you can manage the Monte Carlo results either to show the most relevant data, or to create statistical corners that can be used in the next simulation runs:

- Viewing Data for a Specific Confidence Interval
- Creating Statistical Corners
- Filtering Out Error Data from the Yield View

Viewing Data for a Specific Confidence Interval

During or after a Monte Carlo analysis run, you can specify a confidence level to display the confidence intervals for the estimated yield, mean, and sigma.

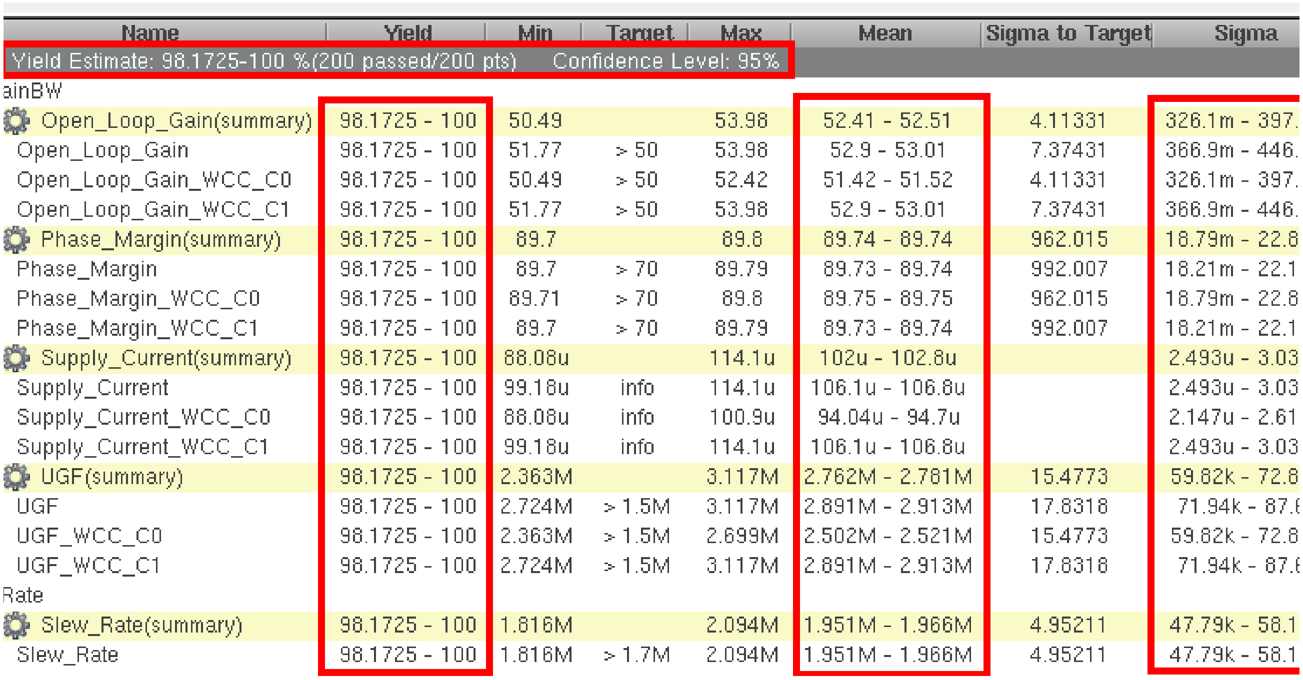
To set the confidence level, right-click the gray-colored row at the top and choose *Set Confidence Level*. The Set Confidence Level form is displayed, as shown in the following figure.

Figure 7-1 The Set Confidence Level Form

Specify a percentage value in the *Confidence Level* field and click *OK*.

ADE XL displays the confidence intervals for the yield, mean, and sigma values, as shown in the following figure.

Figure 7-2 Yield View Displaying Confidence Intervals



Name	Yield	Min	Target	Max	Mean	Sigma to Target	Sigma
Yield Estimate: 98.1725-100 %(200 passed/200 pts) Confidence Level: 95%							
ainBW							
Open_Loop_Gain(summary)	98.1725 - 100	50.49		53.98	52.41 - 52.51	4.11331	326.1m - 397.
Open_Loop_Gain	98.1725 - 100	51.77	> 50	53.98	52.9 - 53.01	7.37431	366.9m - 446.
Open_Loop_Gain_WCC_C0	98.1725 - 100	50.49	> 50	52.42	51.42 - 51.52	4.11331	326.1m - 397.
Open_Loop_Gain_WCC_C1	98.1725 - 100	51.77	> 50	53.98	52.9 - 53.01	7.37431	366.9m - 446.
Phase_Margin(summary)	98.1725 - 100	89.7		89.8	89.74 - 89.74	962.015	18.79m - 22.8
Phase_Margin	98.1725 - 100	89.7	> 70	89.79	89.73 - 89.74	992.007	18.21m - 22.1
Phase_Margin_WCC_C0	98.1725 - 100	89.71	> 70	89.8	89.75 - 89.75	962.015	18.79m - 22.8
Phase_Margin_WCC_C1	98.1725 - 100	89.7	> 70	89.79	89.73 - 89.74	992.007	18.21m - 22.1
Supply_Current(summary)	98.1725 - 100	88.08u		114.1u	102u - 102.8u		2.493u - 3.03
Supply_Current	98.1725 - 100	99.18u	info	114.1u	106.1u - 106.8u		2.493u - 3.03
Supply_Current_WCC_C0	98.1725 - 100	88.08u	info	100.9u	94.04u - 94.7u		2.147u - 2.61
Supply_Current_WCC_C1	98.1725 - 100	99.18u	info	114.1u	106.1u - 106.8u		2.493u - 3.03
UGF(summary)	98.1725 - 100	2.363M		3.117M	2.762M - 2.781M	15.4773	59.82k - 72.8
UGF	98.1725 - 100	2.724M	> 1.5M	3.117M	2.891M - 2.913M	17.8318	71.94k - 87.
UGF_WCC_C0	98.1725 - 100	2.363M	> 1.5M	2.699M	2.502M - 2.521M	15.4773	59.82k - 72.8
UGF_WCC_C1	98.1725 - 100	2.724M	> 1.5M	3.117M	2.891M - 2.913M	17.8318	71.94k - 87.
Rate							
Slew_Rate(summary)	98.1725 - 100	1.816M		2.094M	1.951M - 1.966M	4.95211	47.79k - 58.1
Slew_Rate	98.1725 - 100	1.816M	> 1.7M	2.094M	1.951M - 1.966M	4.95211	47.79k - 58.1

Tip

To revert to the default view without any confidence interval, specify `0` in the *Confidence Level* field.

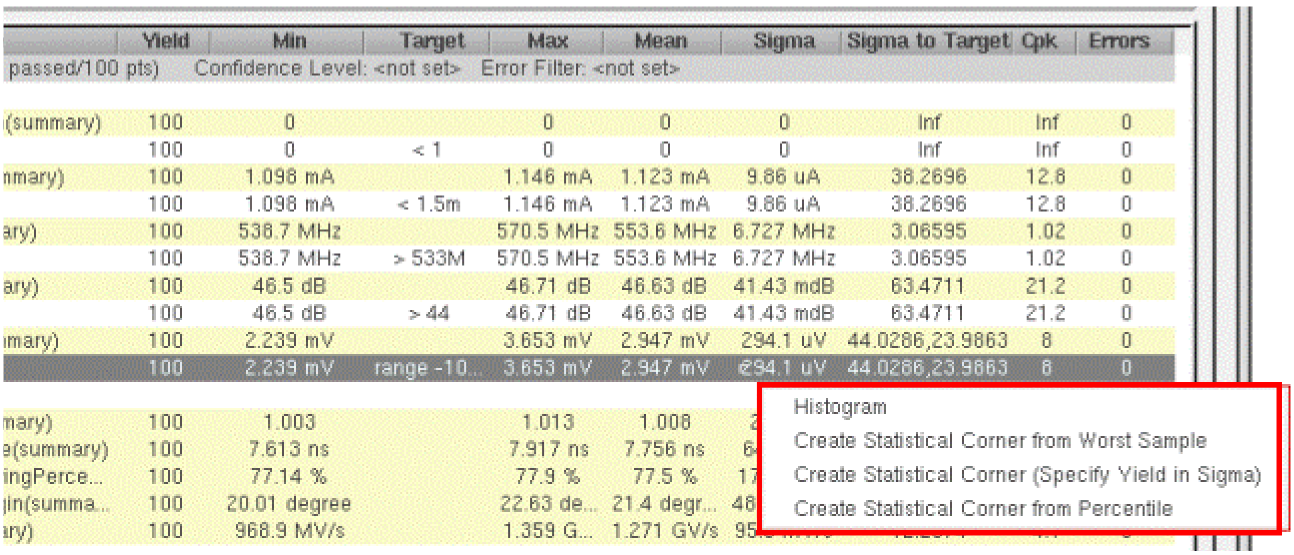
Creating Statistical Corners

After running Monte Carlo simulations, you can analyze the yield and identify the specifications for which the results need improvement. You can then create statistical corners to be used in further analysis and design optimization.

You can create statistical corners by using any of the following four methods:

- From a selected sample
- From a worst sample
- From a percentile
- By using a target yield value (specified in sigma) for a specification

For the first method, you can use the *Create Statistical Corner* command in the right-click pop-up menu of the *Detail* or *Detail – Transpose* results views. The commands for the next three methods are available in the right-click pop-up menu for the specification results in the *Yield* results view, as shown below.



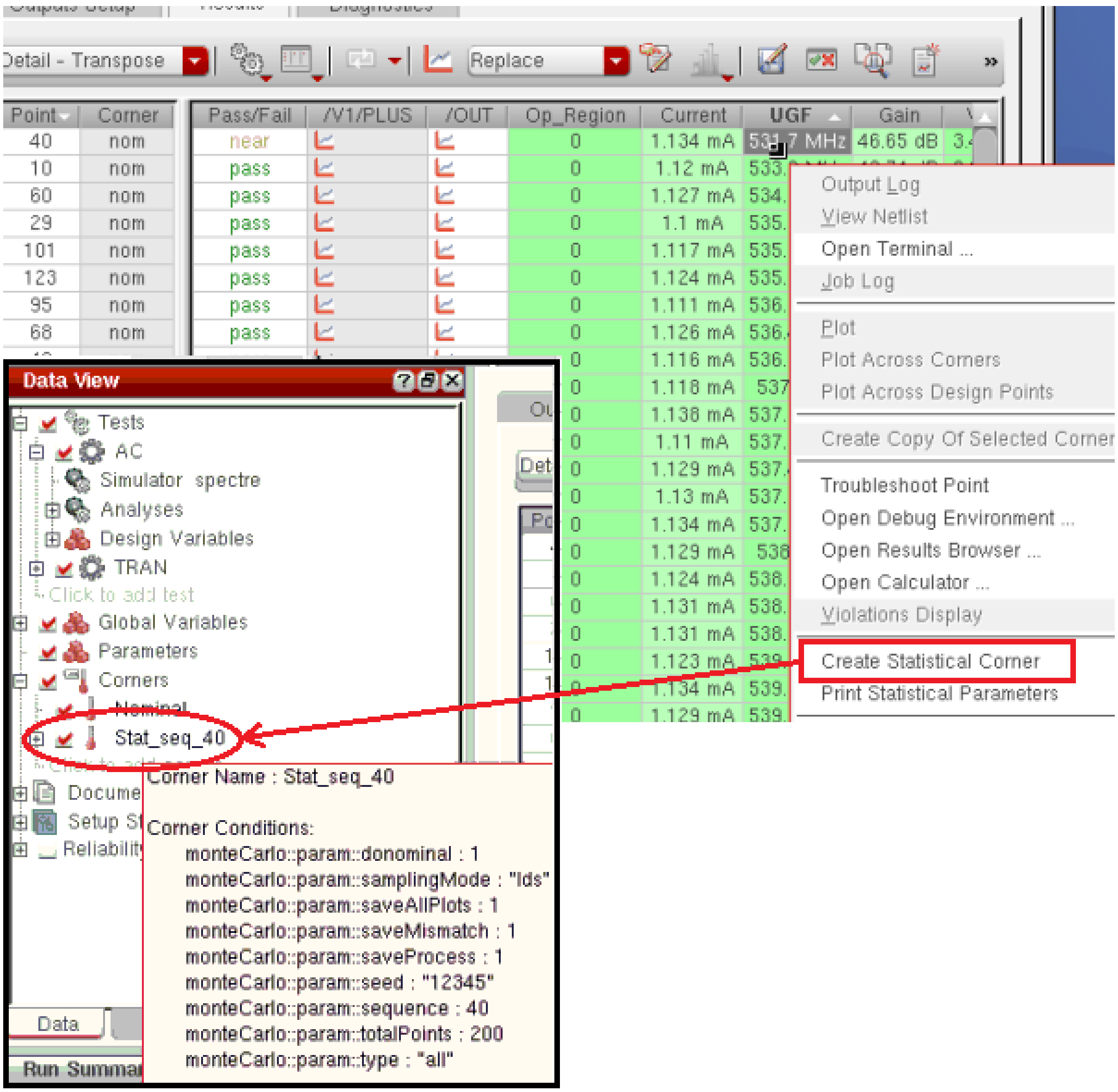
For more details on these ways, refer to the topics given below.

Creating a Statistical Corner from a Selected Sample

You can create a statistical corner from any Monte Carlo sample. For this, do one the following:

- Plot a histogram for the desired specification. In the plot, click a point for which you want to create a corner. ADE XL cross probes the corresponding result in the *Detail* view where you can right-click at the point and choose *Create Statistical Corner*.
- Open the *Detail* or *Detail – Transpose* view. Identify a sample data for which you want to create a corner, right-click on it, and choose *Create Statistical Corner*.

In either of the two ways described above, a statistical corner is created using the sequence ID of the sample. An example is shown in the figure given below.



Creating a Statistical Corner from a Worst Sample

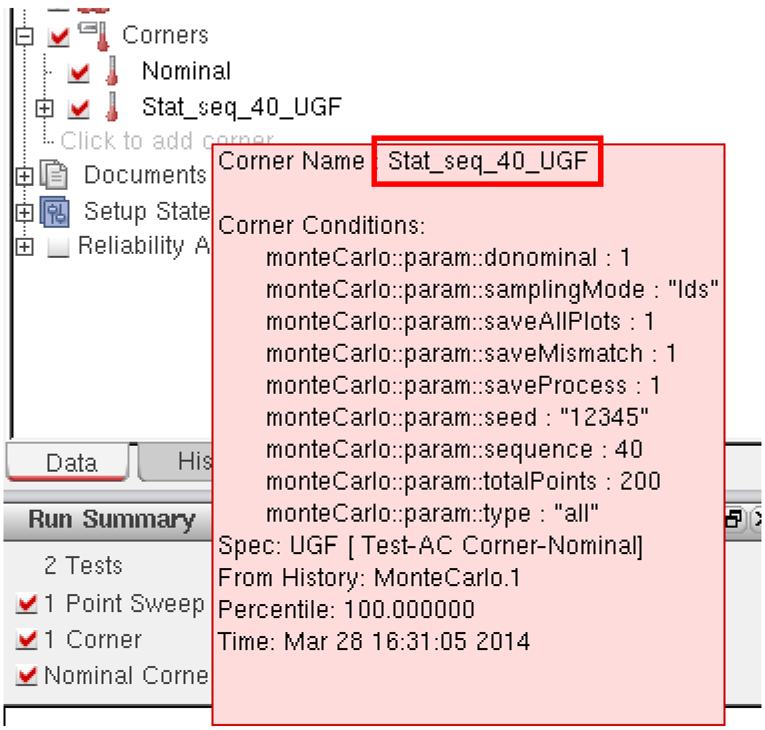
In this method, ADE XL helps in identifying a sample that generated the worst result for a specification and creates a statistical corner using the sample details.

To create a statistical corner from the worst sample, do the following:

1. Open the Monte Carlo results in the *Yield* view and identify the specification for which you need to improve the yield.
2. Right-click on the row for that specification and choose *Create Statistical Corner from Worst Sample*.

ADE XL looks into the simulation results to find a sample that gives the worst result for that specification and creates a statistical corner using that sample. The new corner is added it to the *Corners* list in the Data View assistant pane. The corner is named as Stat_Seq_*<seq_ID>*_*<spec_name>*, where <seq_ID> is the sequence ID of that sample and *<spec_name>* is the name of the specification. For example, if you choose to create a corner for the worst sample for `UGF`, ADE XL creates a corner with the name `Stat_seq_40_UGF`, where the sequence ID of the sample is 40.

Note the details of the statistical corner in the following figure.

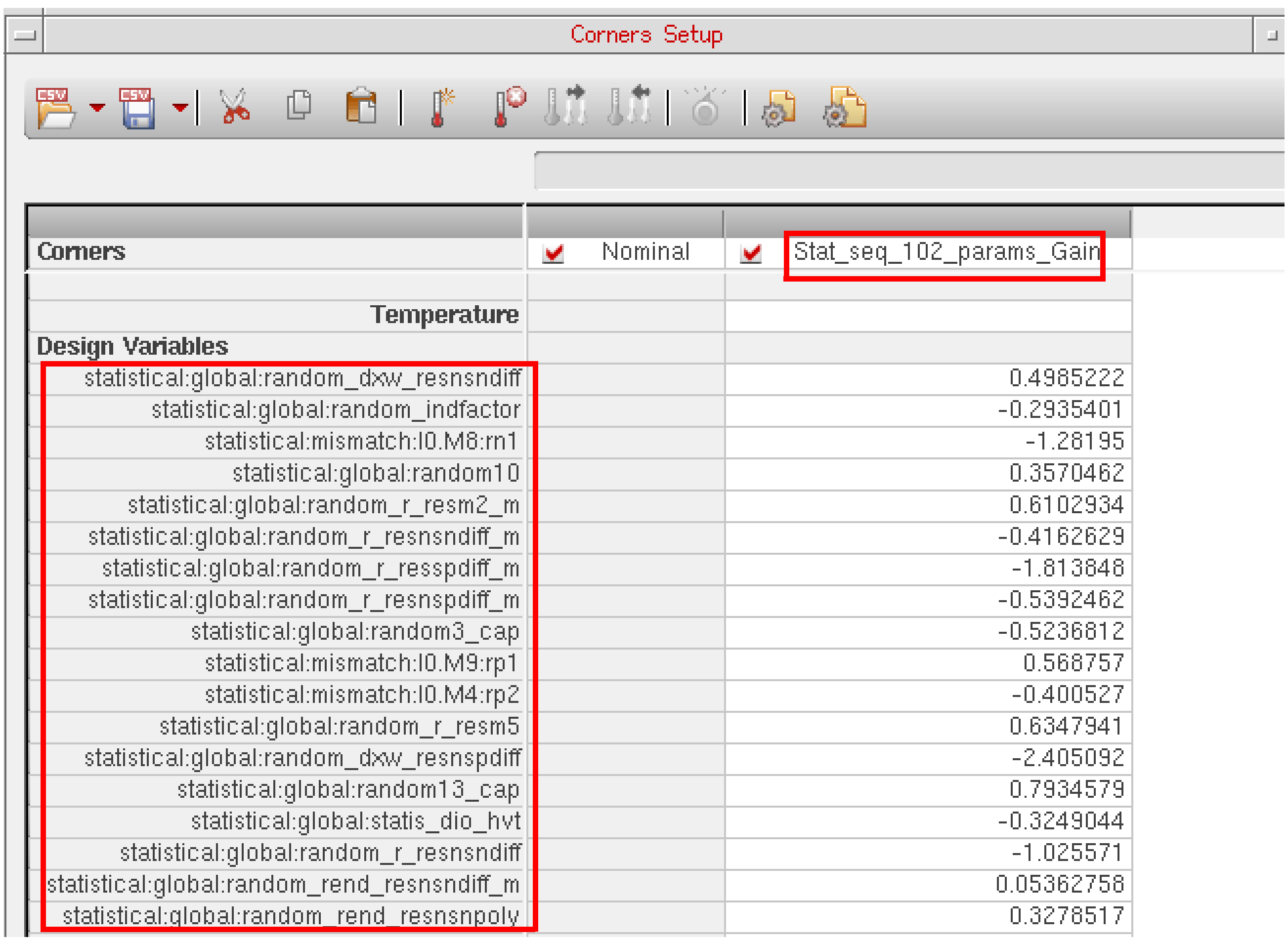


The figure given above shows that ADE XL saves the sequence ID, seed, and other Monte Carlo settings to the statistical corner when simulating it. Using this information, the simulator can recreate the statistical parameter values for the corner. This type of corner is efficient and does not require saving all the statistical parameter data for each sample in Monte Carlo. However, one disadvantage is that some changes to the design topology, such as, addition of a new instance to the schematic, can invalidate the corner. In that case, the simulator cannot recreate the same set of statistical parameter values when simulating the statistical corner. You would then need to rerun the Monte Carlo simulation to recreate statistical corners.

To avoid this, you can create statistical corners by directly saving the parameter values from the sample in the corner details and by not referring to any sequence ID. For more details, refer to Creating a Statistical Corner with Statistical Parameter Values.

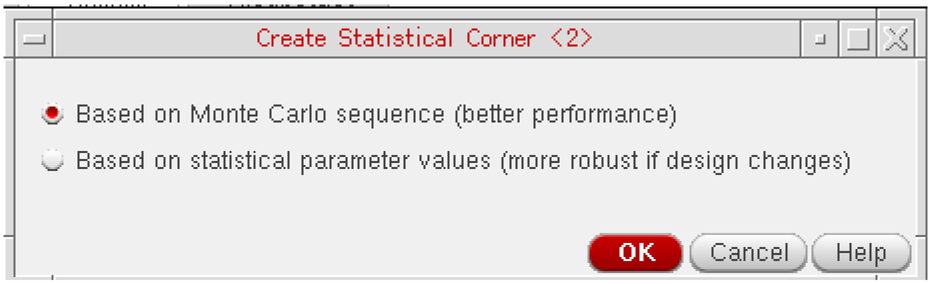
Creating a Statistical Corner with Statistical Parameter Values

You can create a statistical corner by directly saving the parameter values from the sample in the corner details. These corners are named as `Stat_Seq_`*`<seq_ID>`*`_params` *`<spec_name>`*, where `<seq_ID>` is the sequence ID of that sample and *`<spec_name>`* is the name of the specification. The corner details in the Corner Setup form show all the statistical parameters and their values set for the corner, as shown in the figure given below.



To create statistical corners by saving the parameter values, you need to change the default value of the createStatisticalCornerType environment variable. By default, this variable is set to `sequence`, which indicates that the statistical corners are to be created by using the sequence number of a sample from simulation results. Set this environment variable to `values` to create the corners by using the statistical parameter values.

If you wan to choose a way to create a statistical corner at the runtime, set the environment variable either to `prompt` or `promptValues`. Now, when you right-click on a specification in the *Yield* view and choose *Create Statistical Corner from Worst Sample*, ADE XL displays the Create Statistical Corner form.



If the createStatisticalCornerType environment variable is set to `prompt`, the default selection in the message box is `Based on Monte Carlo sequence`. You can change the default selection by setting the variable to `promptValues`, which makes `Based on statistical parameter values` as the default value.

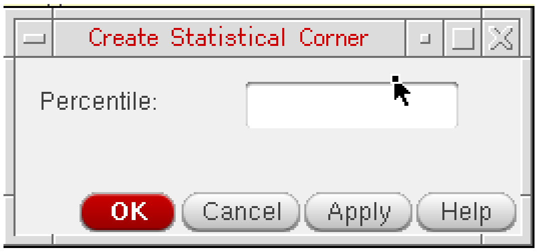
Important

Statistical corners created in this way are more robust and do not need to be recreated if there are minor design changes. However, in this case, you need to save the statistical parameters while running the Monte Carlo simulations.

Creating a Statistical Corner from Percentile

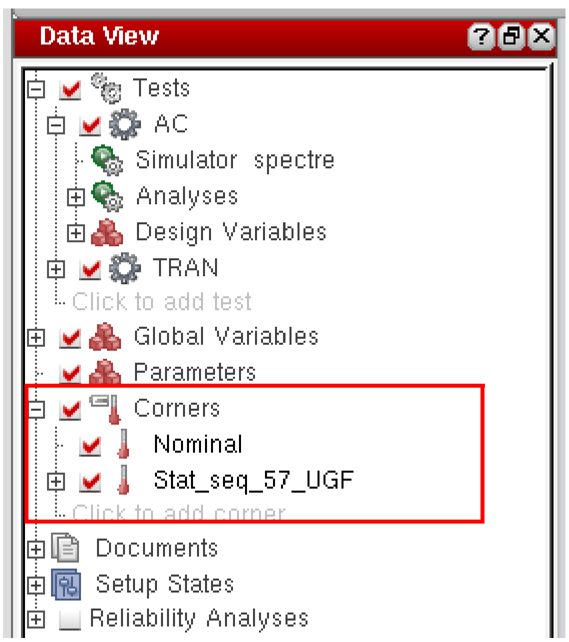
In this method, you can create a corner for the nth worst percentile for a given specification. That is, a statistical corner is created from the Monte Carlo sample that gives the nth worst output for a specification.

To create a statistical corner from percentile, right-click the output of a specification in the *Yield* view and choose *Create Statistical Corner from Percentile*. The Create Statistical Corner form is displayed, as shown in the following figure.

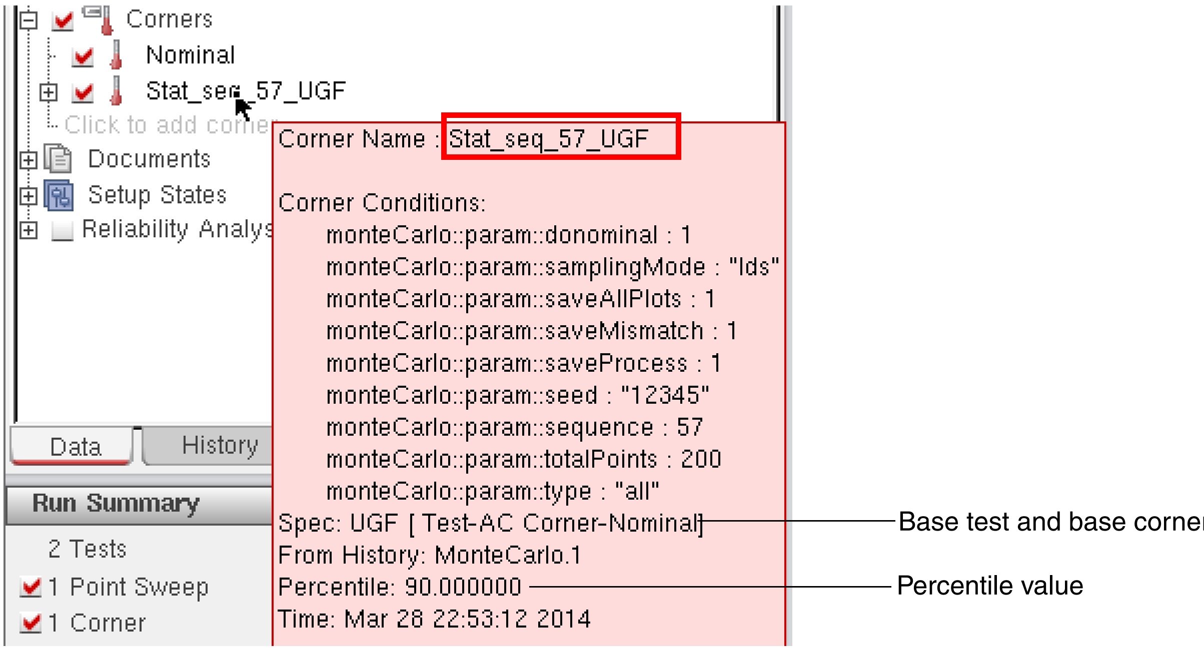


Specify a percentile value for which you want to create a sample and click *OK*.

ADE XL creates a new statistical corner and adds it to the Corners node in the Data View assistant pane, as shown in the following figure.



The default name of a new statistical corner consists of a prefix `Stat_seq` followed by the sequence ID of the sample and the specification name. The tooltip for a corner displays the values for all corner conditions, names of the base specification and base corner, and the percentile value used to create that corner, as shown in the following figure.



Important

If the specification type is < or >, only one statistical corner is created from the percentile. If the specification type is `range` or if a specification is not defined, two corners are created–one in each direction.

By default, the corner uses the sequence ID of the sample that gives the nth worst output for a specification. Using this information, the simulator can then recreate the statistical parameter values for the corner when simulating it. This type of corner is efficient and does not require saving all the statistical parameter data for each sample in Monte Carlo. However, one disadvantage is that some changes to the design topology, such as, addition of a new instance to the schematic, can invalidate the corner. In that case, the simulator cannot recreate the same set of statistical parameter values when simulating the statistical corner. You would then need to rerun the Monte Carlo simulation to recreate statistical corners.

To avoid this, you can create statistical corners by directly saving the parameter values from the sample in the corner details and by not referring to any sequence ID. For more details, refer to Creating a Statistical Corner with Statistical Parameter Values.

Important

A statistical corner is enabled only for the test associated with the specification for which the corner is created.

Creating Statistical Corners by Using Target Yield Value (Specified in Sigma)

This is a fast method to create 3-sigma or user-specified sigma statistical corners without requiring a large number of Monte Carlo samples.

Note: Creating statistical parameters by this method requires the `Analog_Design_Environment_GXL` license. For more details, refer to Creating a K-Sigma Statistical Corner in *Analog Design Environment GXL User Guide*.

Filtering Out Error Data from the Yield View

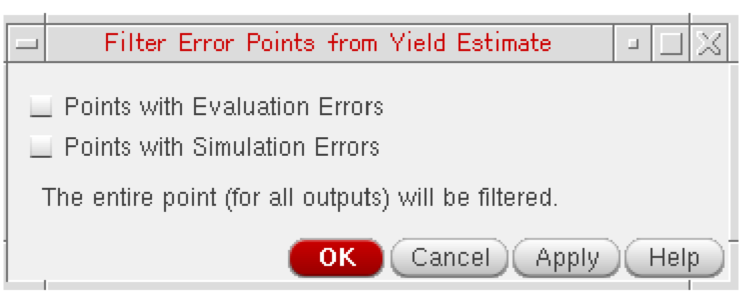
By default, the statistical calculations displayed in the *Yield* view are based on all the points in Monte Carlo results. However, you can filter out and eliminate the points that resulted in simulation or evaluation errors so that the yield estimate is calculated only on the points for which simulation was successful.

Important

A typical Monte Carlo error point represents a real failure of the design. Therefore, it is recommended to filter the error points only when it is certain that the error points are invalid.

To apply a filter in the *Yield* view, right-click the gray-colored row at the top and choose *Filter Error Points from Yield Estimate*. The Filter Error Points from Yield Estimate form is displayed, as shown in the following figure.

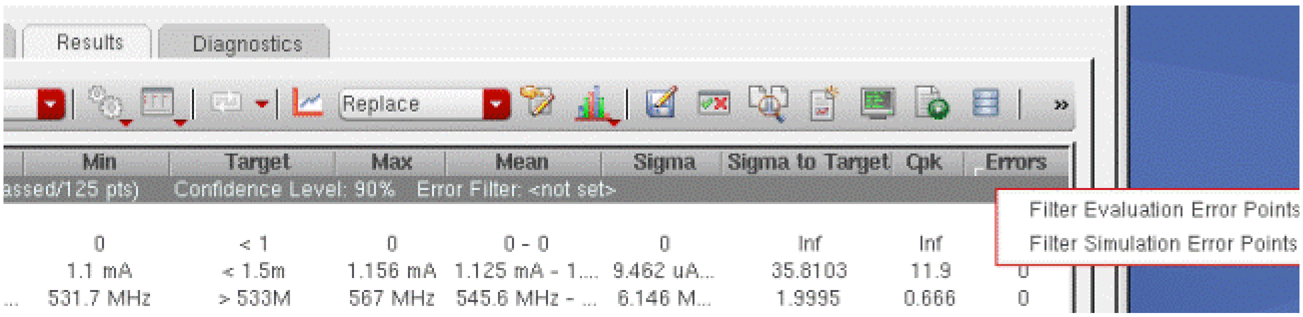
Figure 7-3 The Filter Error Points from Yield Estimate Form



Select a check box to filter the required type of errors and click *OK*.

Alternatively, you can right-click on the *Errors* column title in the *Yield* view and choose the type of filter to be applied, as shown in the following figure.

Figure 7-4 The Context-Sensitive Menu of the Errors Column



ADE XL filters out the error points from the *Yield* view. The yield values for specifications and the overall yield estimate are updated. Note that the total number of points considered for yield estimate calculation are also reduced to eliminate the points with errors, as shown in the figures given below.

Figure 7-5 Yield View Before Applying the Error Filter

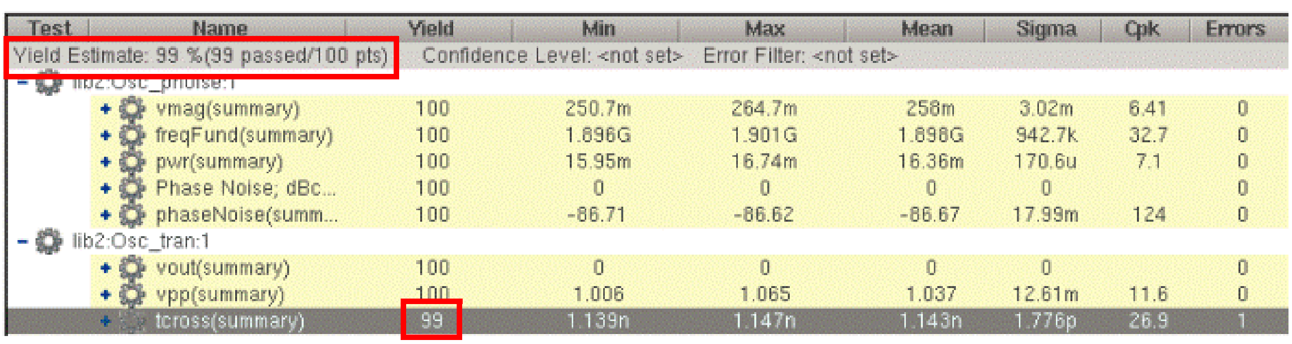
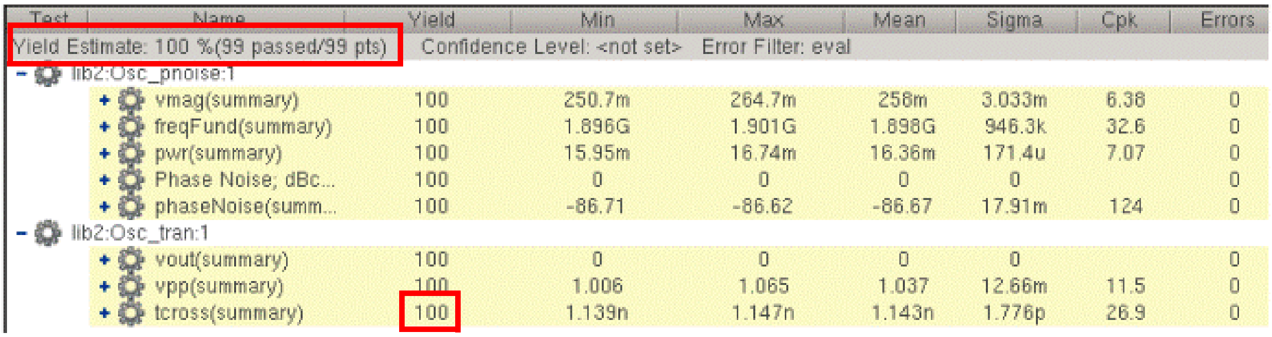


Figure 7-6 Yield View After Applying Error Filter



Test	Name	Yield	Min	Max	Mean	Sigma	Cpk	Errors
Yield Estimate: 100 %(99 passed/99 pts)		Confidence Level: <not set>		Error Filter: eval				
lib2:Osc_pnoise:1								
	vmag(summary)	100	250.7m	264.7m	258m	3.033m	6.38	0
	freqFund(summary)	100	1.896G	1.901G	1.898G	946.3k	32.6	0
	pwr(summary)	100	15.95m	16.74m	16.36m	171.4u	7.07	0
	Phase Noise; dBc...	100	0	0	0	0		0
	phaseNoise(summ...	100	-86.71	-86.62	-86.67	17.91m	124	0
lib2:Osc_tran:1								
	vout(summary)	100	0	0	0	0		0
	vpp(summary)	100	1.006	1.065	1.037	12.66m	11.5	0
	tcross(summary)	100	1.139n	1.147n	1.143n	1.776p	26.9	0

Important Points to Note

- The error filter only filters the results displayed in the *Yield* view and does not remove the results from the results database. At any time, you can remove the filter to view the results for all the points.
- If there are multiple tests, a point that is filtered from the output statistics of a particular test is also filtered out from the statistics of other tests.

Generating Plots, Tables, and Reports

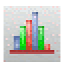
After a Monte Carlo run, you can generate plots, tables, and reports for your input and output parameters. See the following topics for more information:

- Plot/Print versus Iteration on page 311
- Printing Correlation Tables on page 314
- Plotting Histograms on page 316
- Plotting Scatter Plots on page 324

Plot/Print versus Iteration

Note: Starting from IC6.1.5 ISR15, the *Print Iteration vs. Value* command has been changed to *Plot/Print versus Iteration*. You can now use this command to print as well as plot the values of outputs and parameters. In addition, you can also plot or print the mean and sigma values for outputs.

To plot or print the values of outputs or parameters at the end of each iteration, do one of the following:

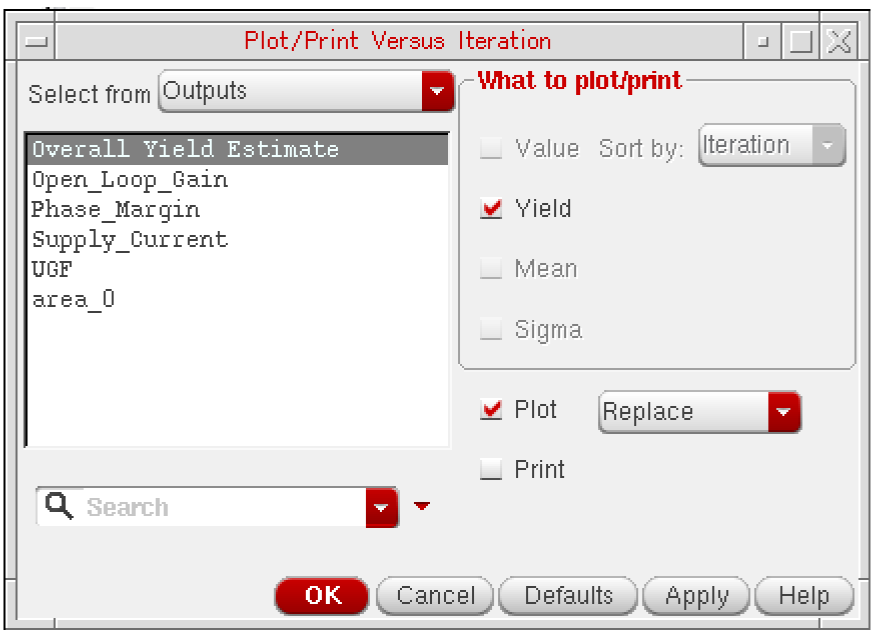
1. Click  on the Results tab, select the test name and choose *Plot/Print Versus Iteration.*

   Alternatively, do the following:

   a. On the Results tab of the Outputs pane, switch to the *Yield* view.

   b. Right-click a test name and choose *Print Iteration vs. Value*.

   The Plot/Print Versus Iteration form is displayed.



2. From the *Select from* drop-down list, select the type of results for which you want to plot or print values.

   The default value is `Outputs`. When `Outputs` is selected, the field beneath this list box shows the names of all the outputs. In addition, the `Overall Yield Estimate` is also listed on top.

When you select `Statistical Parameters` from the *Select from* drop-down list, the names of all the parameters are displayed in the field beneath the list box.

Note: *Overall Yield Estimate* is useful in plotting the yield estimate graphs for a Monte Carlo simulation.

3. From the scroll list given below the *Select from* list, select the names of one or more output or parameters for which you want to print or plot the result values at the end of each iteration.

4. In the *What to plot/print* section, select appropriate check boxes to specify the types of values to be plotted or printed.

   You can choose to print only the result values or the calculated yield, mean or sigma values for each iteration.

5. In the *Sort by* list, specify the sorting criteria.

   You can sort by iteration number or by the result values. The printed results are sorted in the ascending order.

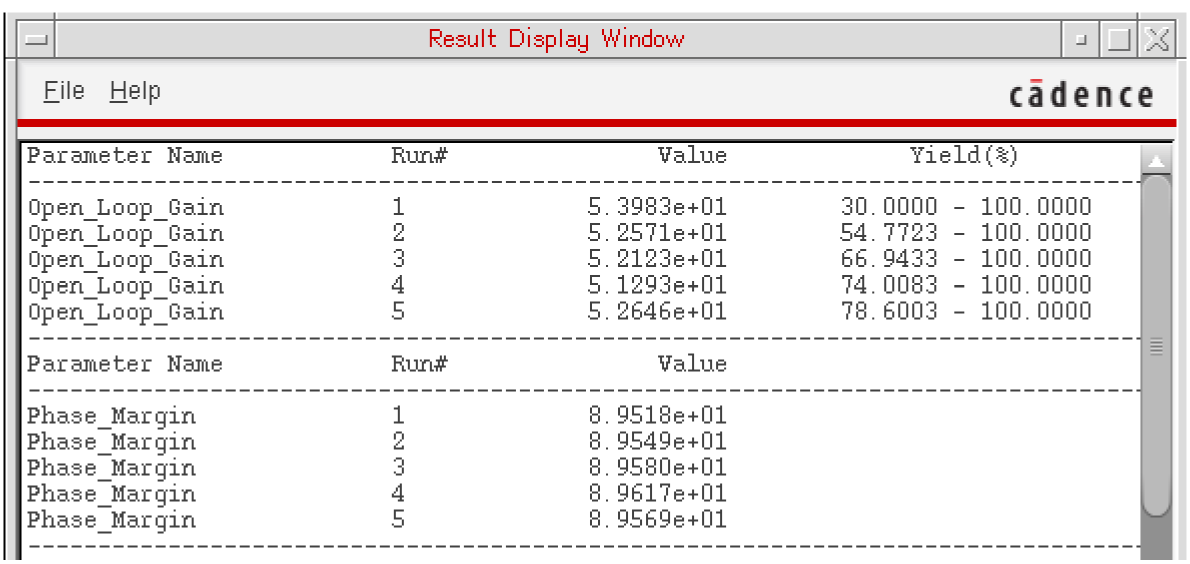
6. Select the *Plot* check box to plot the results.

   You can also choose a plotting mode from the drop-down list given to the right.

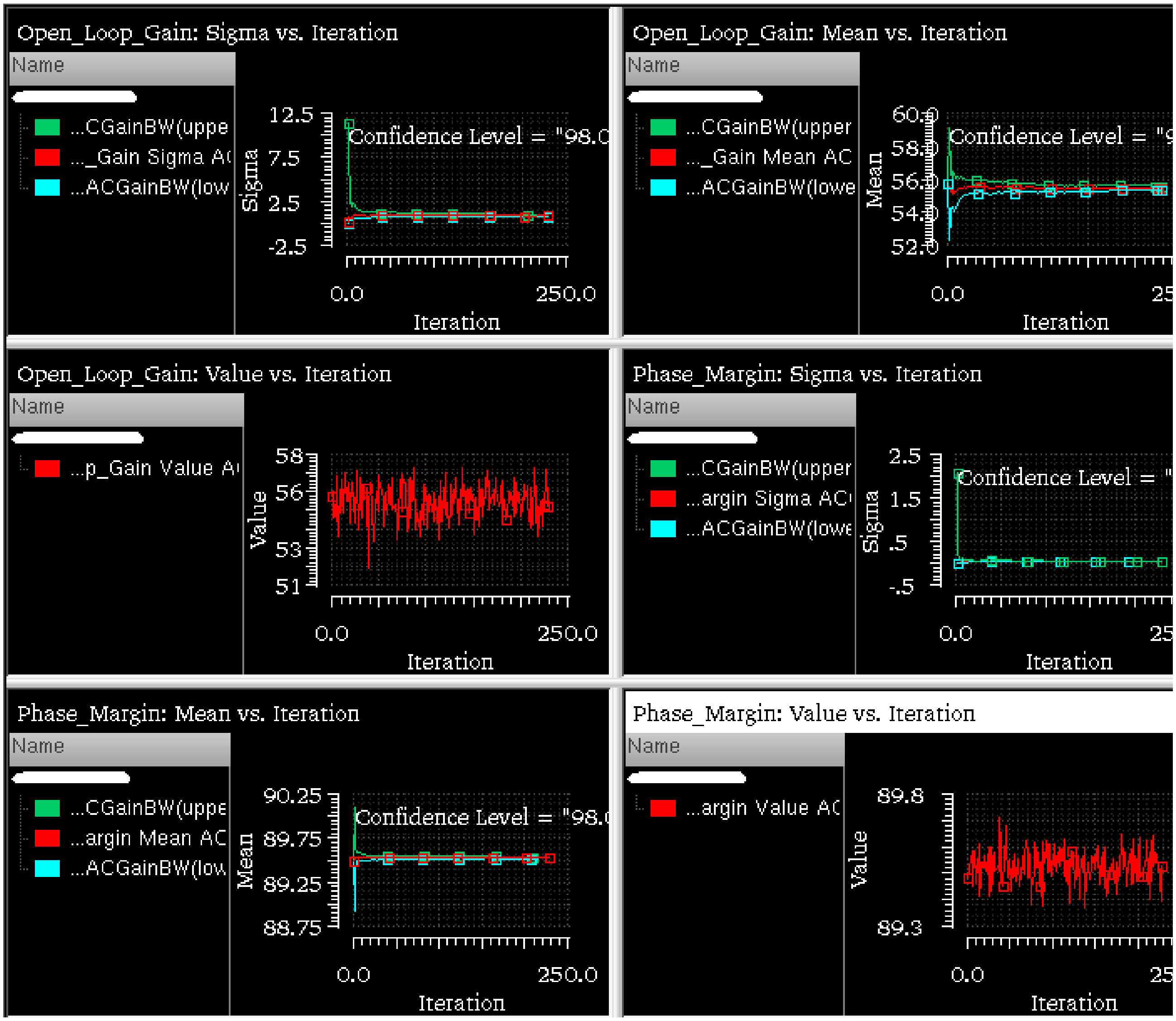
7. Select the *Print* check box to print the results.

8. Click *Apply* to print or plot as per the specifications given on this form.

   The iteration versus value table for the selected outputs or parameters are displayed in a Results Display Window, as shown below.



The plots of the selected options are displayed in the Virtuoso Visualization and Analysis XL window as shown below.



Note: You can incrementally change the options on the Plot/Print Versus Iterations form and print parameters and outputs. In that case, the incremented values are appended to the end of the Results Display Window. Extra columns are also added to the right.

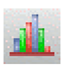
The plotted results are displayed in the Virtuoso Visualization and Analysis XL window. Incrementally displayed plots are shown in the same or new window or subwindow as per the specified plotting mode.

Important Points to Note

- Only the yield value can be printed or plotted for the `Overall Yield Estimate` output. You cannot print or plot mean or sigma value for this output type.
- The *Sort by* field is applicable for printing and plotting of value, mean and sigma.
- You can use the *Search* field to search for a particular output or parameter name.

Printing Correlation Tables

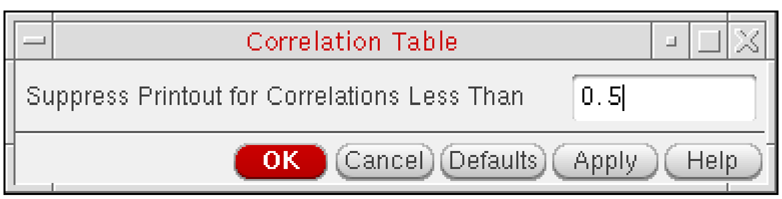
To print a table showing the correlation coefficients of each parameter with each of the other parameters sorted from most correlated to least correlated for each combination of parameters, do one of the following:

- Click  on the Results tab, select a test and choose *Print Correlation.*
- Do the following:

  a. On the Results tab of the Outputs pane, switch to the *Yield* view.

  b. Right-click a test name and choose *Print Correlation*.

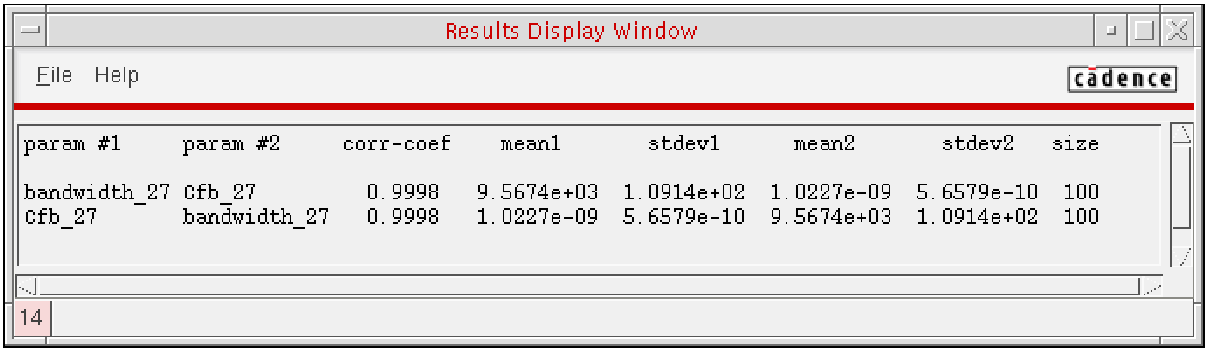
  The Correlation Table form appears.



1. In the *Suppress Printout for Correlations Less Than* field, type a minimum correlation value.

2. Click *OK*.

Pairs of parameters with correlation coefficients greater than the value you typed in the *Suppress Printout for Correlations Less Than* field appear in a table in a Results Display Window.

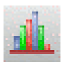


Each row lists the pair of measurements considered, the mean and standard deviation of the first measurement, the mean and standard deviation of the second measurement, and the number of data points included in the calculation.

Plotting Histograms

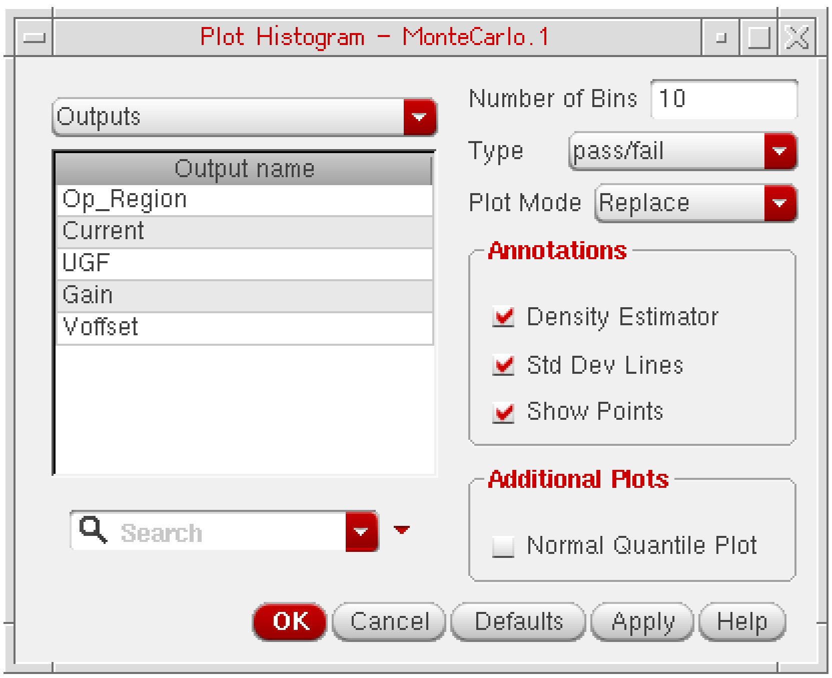
You can plot histograms for the outputs and statistical parameters from Monte Carlo results.

To plot histograms, do the following:

1. On the Results tab of the Outputs pane, switch to the *Yield* view.
2. Click  on the Results tab, select a test and choose *Histogram.*

   Alternatively, right-click a test name and choose *Histogram*.

   The Plot Histogram form appears.

   By default, `Outputs` is selected from the drop-down list on the top-left corner of the form and a list of names of all the outputs (from the *Outputs Setup* tab) is also listed in the *Output name* list.

3. Select one or more output names that you want to plot on the histogram.

   Note: You can drag or *Shift*-click to select a group of adjacent outputs or *Control*-click to select individual outputs.

   Alternatively, you can choose to plot histograms for statistical parameters. For this, select `Statistical Parameters` from the drop-down list on the top-left corner of the form and select the names of one or more parameters from the list below it.

Tip

If the list of outputs or statistical parameters is very long, you can use the search field to search for a specific output or parameter name.

4. Specify a value from 1 to 50 in the *Number of Bins* field.

   The default value is `10`.

5. Select an appropriate histogram graph type from the *Type* drop-down list. You can select one of the following:
   - ❑ *standard*
   - ❑ *pass/fail*
   - ❑ *cumulative line*
   - ❑ *cumulative box*

6. (Optional) Specify a plotting mode to specify how to plot graphs when a Virtuoso Visualization and Analysis XL window is already open.

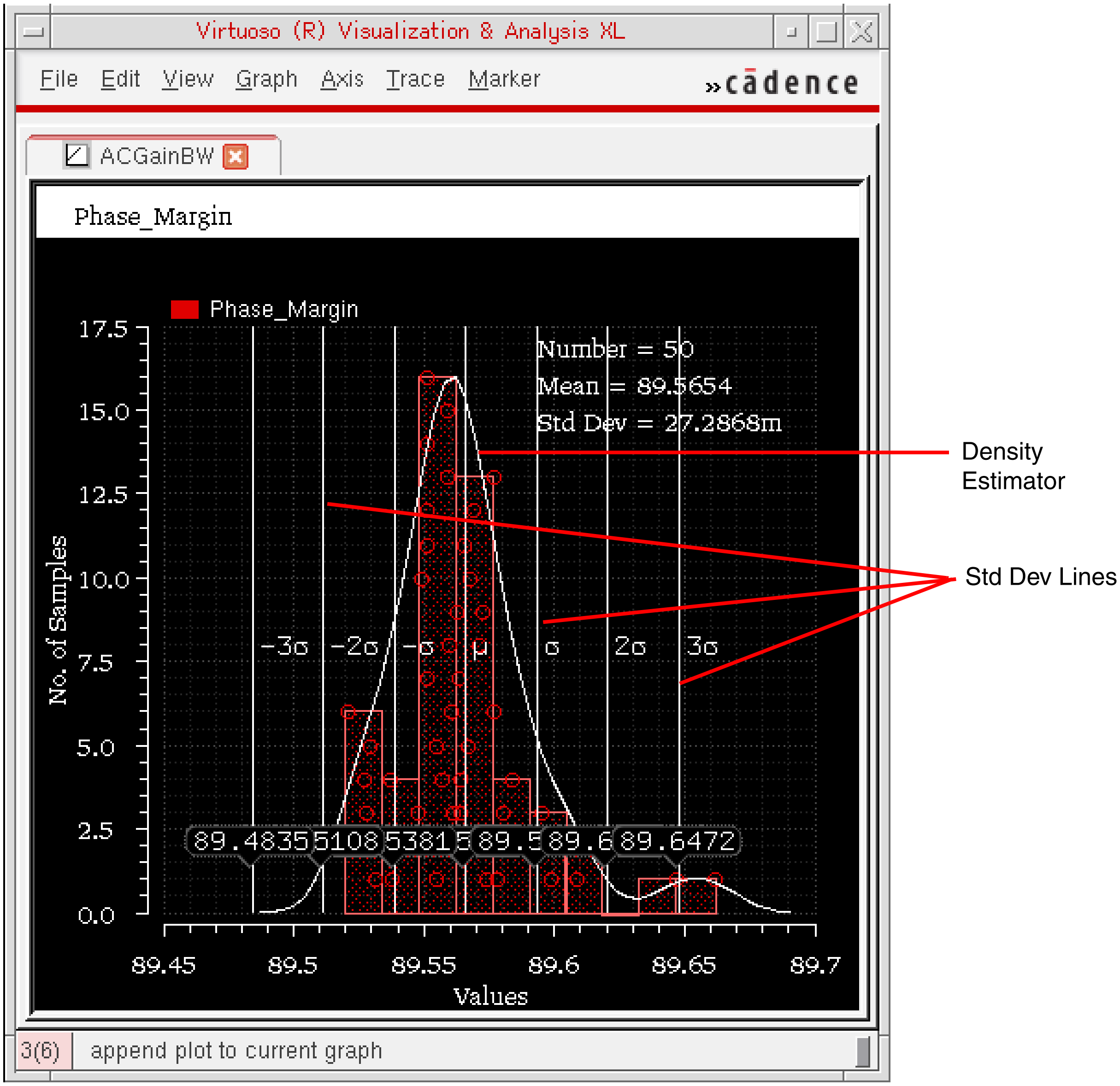
   The default plotting mode is `Replace`. For more details on the plotting modes, refer to Selecting the Plot Mode.

7. (Optional) Select the types of annotations that you want to display on the histogram plots. You can select one of the following annotation types:
   - ❑ *Density Estimator*: Plots a curve that estimates the distribution concentration.
   - ❑ *Std Dev Lines*: Shows the standard deviation lines in the graph indicating the mean, mean – standard deviation, and mean + standard deviation values.

8. (Optional) Select the *Normal Quantile Plots* check box to plot the quantile plots (Q-Q plots).

9. Click *OK*.

The histogram plot appears in the Virtuoso Visualization and Analysis XL window, as shown below.



The standard deviation lines indicate how the data points are spread, far away or close to the mean values. The gap between each vertical line is equal to one standard deviation.

Important Points to Note

- The default histogram type is `pass/fail`. You can choose a different plot type from the *Type* drop-down list on the Plot Histogram form.

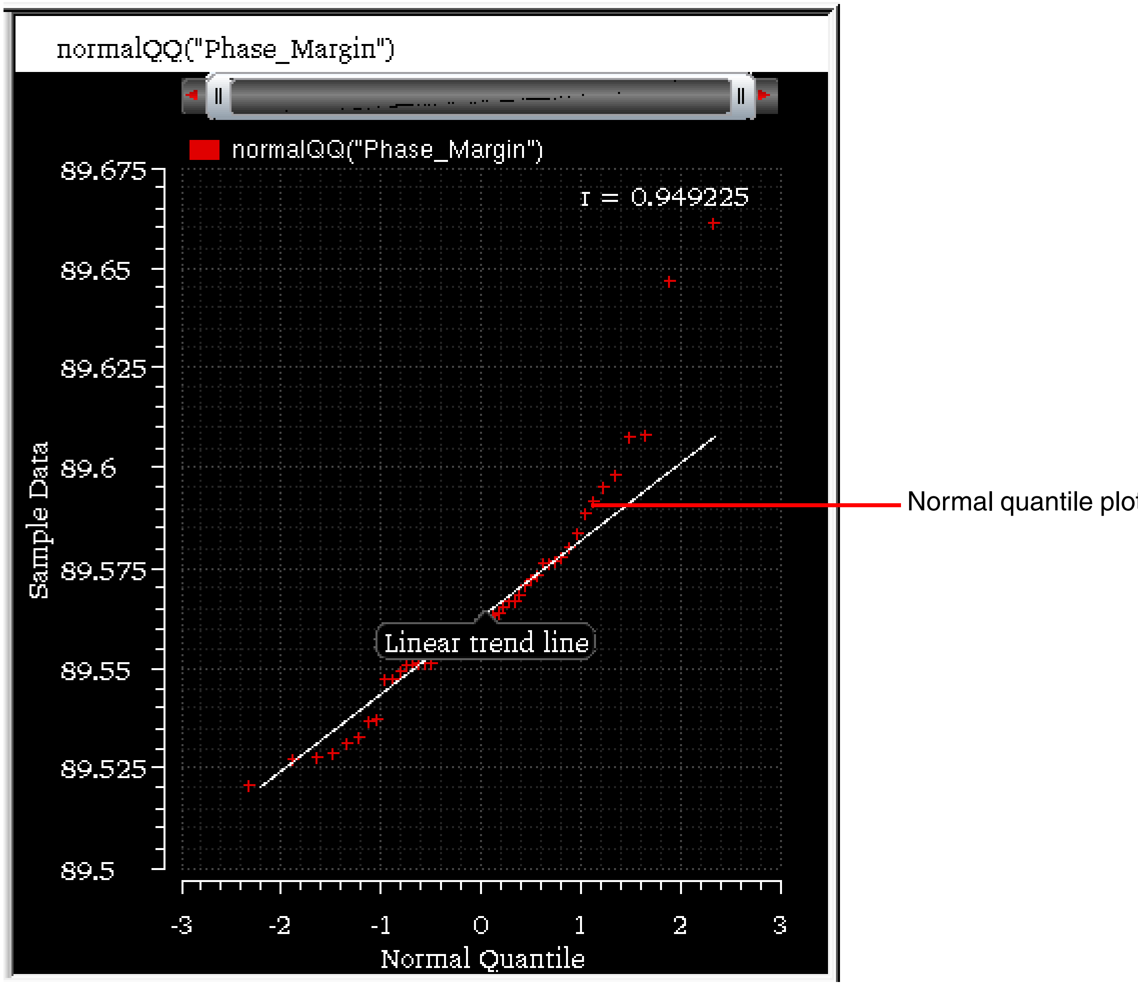
- The annotations displayed in the graph vary depending on the histogram type.
- If you select multiple outputs or parameters, each output or parameter is plotted in a separate subwindow.
- By default, the histogram bars are filled with a transparent or alpha color to make the data points clearly visible. You can click on these points to cross-probe results on the Results tab. For more details, refer to Cross-Probing ADE XL Results from Histogram Plots.
- When you choose to display the normal quantile plot, ADE XL plots the graphs in two subwindows, one shows the histogram and the other shows the normal quantile plot. For details on a normal quantile plot, refer to Normal Quantile Plots.
- If you observe a large variation in the results for a particular output, you can view the Mismatch Contribution table to identify the devices that are contributing to that variation. If required, you can modify such devices in your design and rerun Monte Carlo to verify that the variation is reduced. For details on the how to view data in the Mismatch Contribution table, refer to Analyzing the Mismatch Contribution.
- You can also set the histogram options using the ADE XL Printing/Plotting Options form.
- You can set the default values for the histogram options using the following environment variables:
  - histogramBins
  - histogramType
  - showHistogramDensity
  - showHistogramDeviation
  - showHistogramPoints
  - histogramBins

Important

To view the Mismatch Contribution table, you would require the Analog_Design_Environment_GXL (95220) license.

Normal Quantile Plots

A normal quantile plot (Q-Q plot) helps in comparing the actual result values for an output with the values predicted for a standard normal distribution.



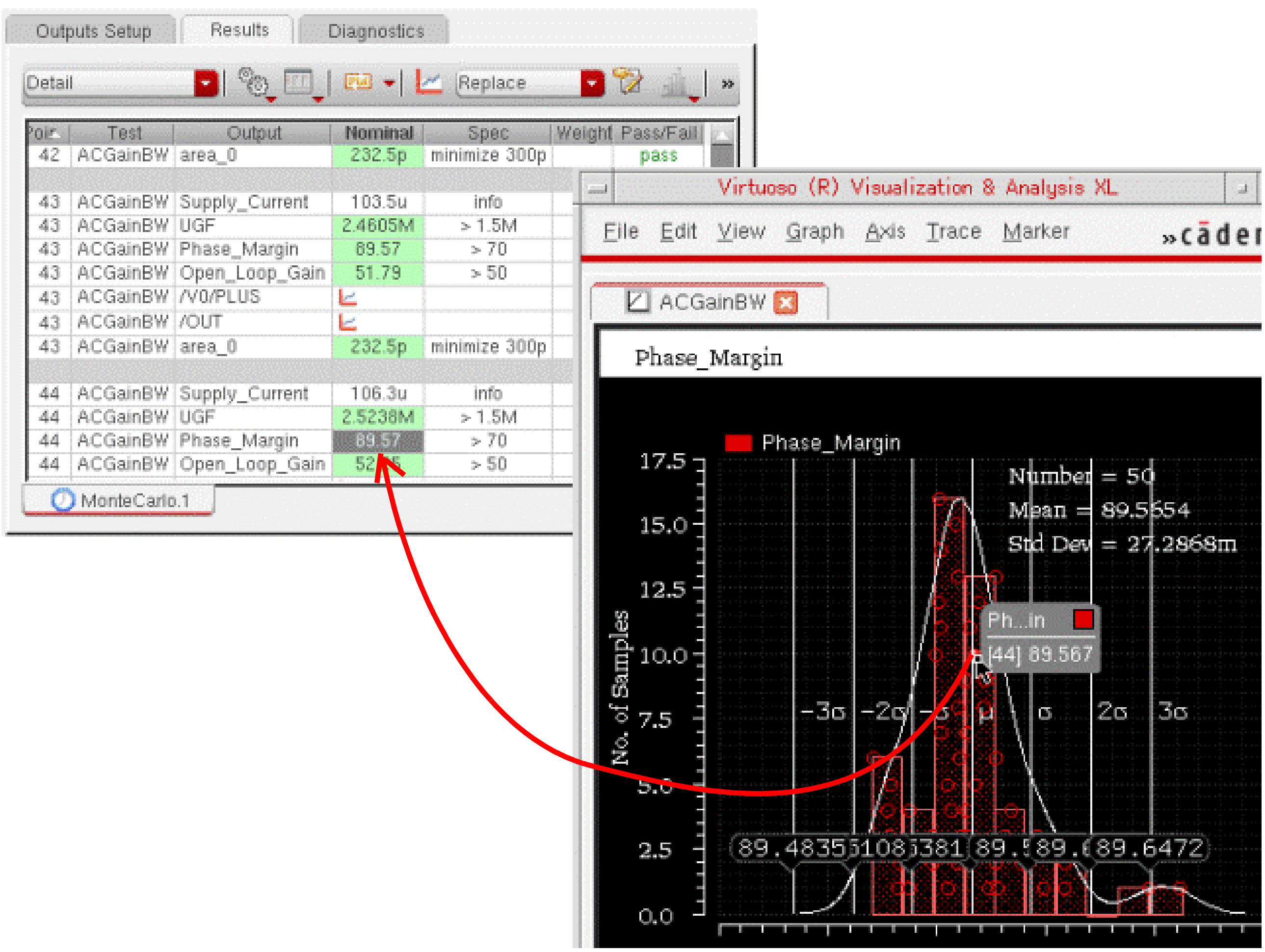
In the plot shown above, Y-axis shows the Monte Carlo results for an output (sample data) and the X-axis shows the normal quantiles of that sample data, which is calculated with `mean=0` and `standard deviation=1`.

You can compare this normal quantile graph (shown in red) with the linear trend line, which is a reference line that signifies how the data is plotted if it is perfectly normally distributed. The correlation between the normal quantiles and the sample data is a good measure to check how well the data is modeled by a normal distribution. Points of a normal quantile graph that fall on the linear trend line indicate a higher positive correlation. Points close to this line indicate that the data is close to normal distribution and the points far from this line indicate that the data is not close to normal distribution.

The correlation coefficient (r) of the points in the plot is calculated and displayed in the legend. This value indicates the relationship of the two variables, sample data and the normal quantiles. This value ranges between -1 and 1, where the value of 1 indicates a perfect positive correlation, -1 indicates a perfect negative correlation, and 0 indicates no correlation between the two.

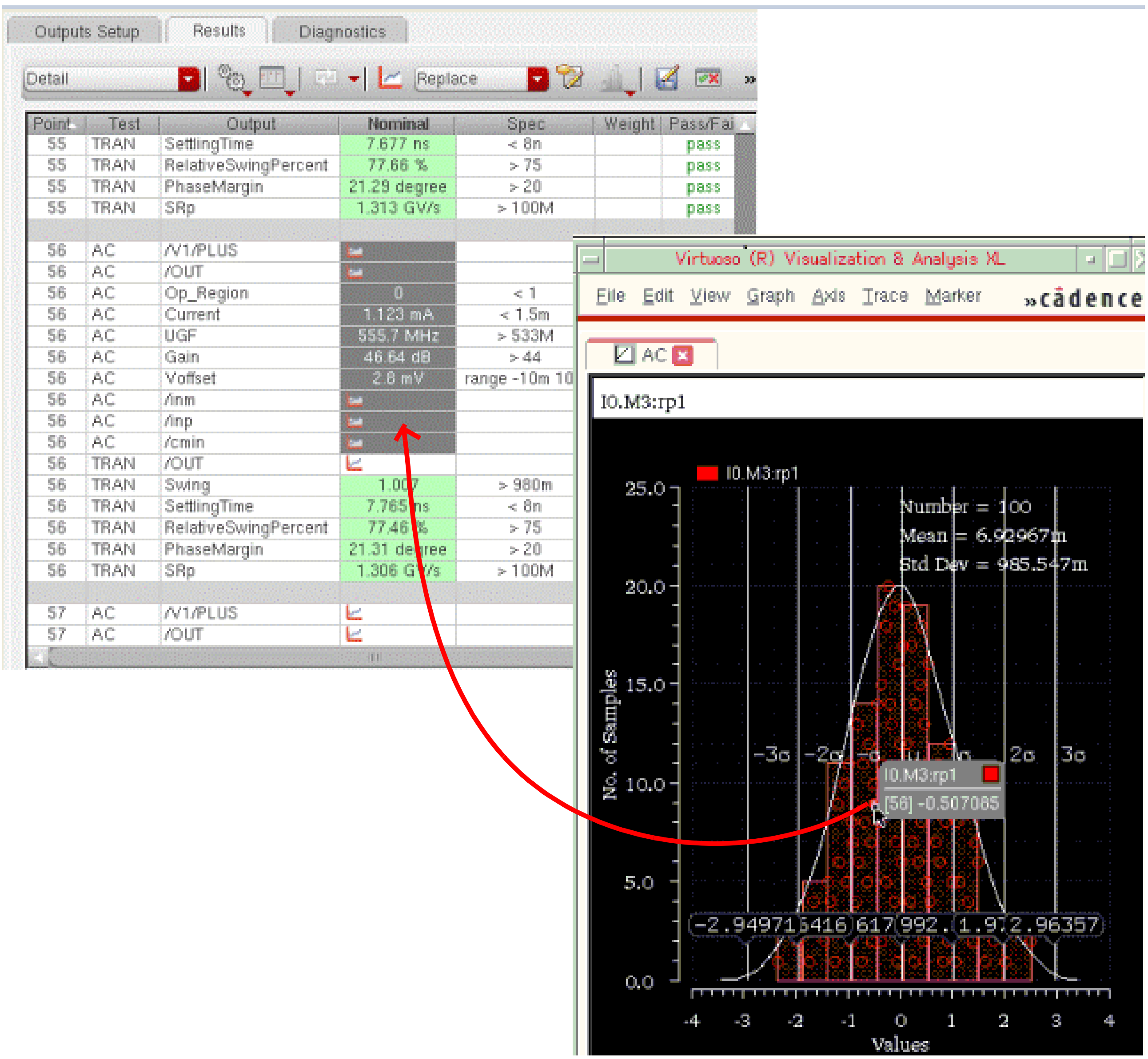
Cross-Probing ADE XL Results from Histogram Plots

By default, the histogram bars are filled with a transparent or alpha color to make the data points clearly visible. When you click on any point on the histogram, the Detail view of results is opened in the *Results* tab and the result corresponding to that point is selected, as shown below.



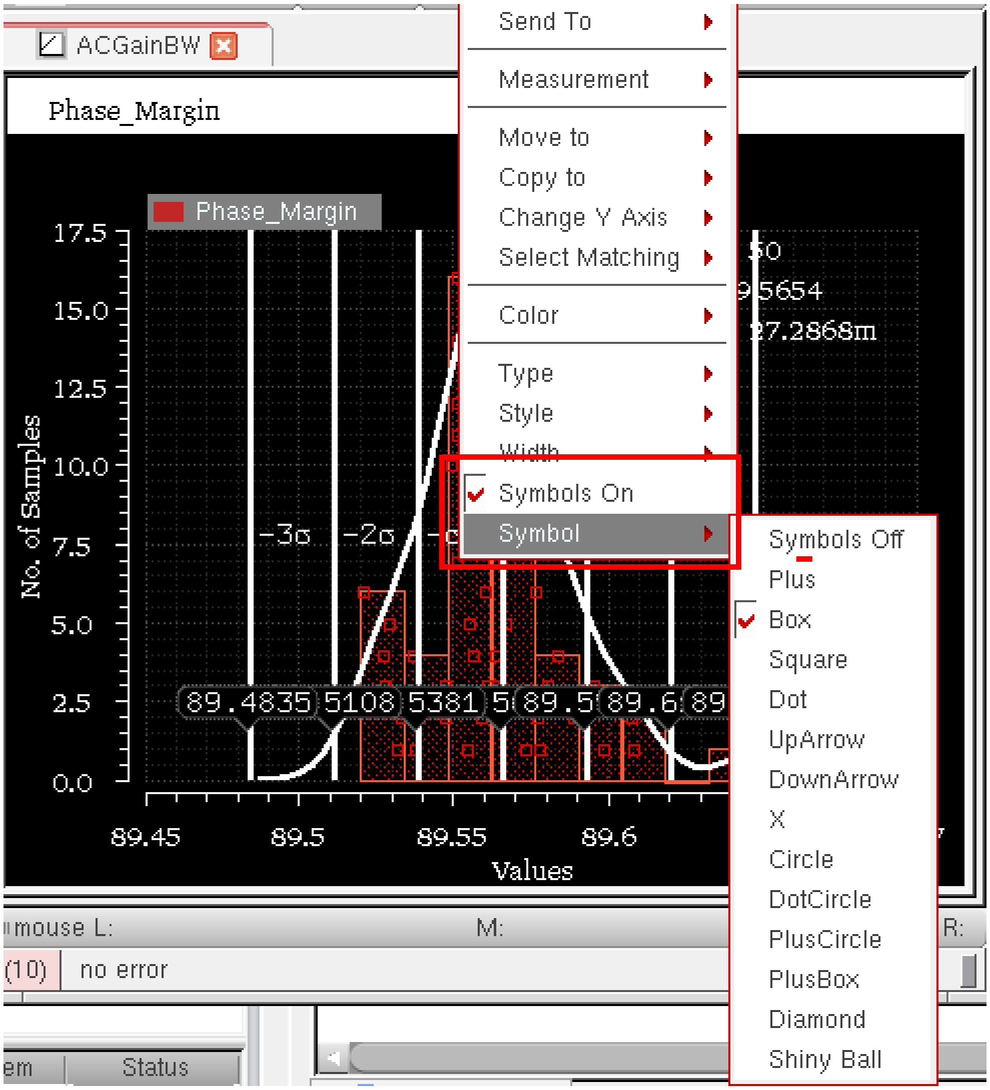
Similarly, you can cross-probe a result value from the Results tab to the histogram.

Cross-probing is supported for the histograms plotted for outputs as well as statistical parameters. When you select a point on the histogram plotted for a statistical parameter, all the outputs of the corresponding design point are selected in the *Detail* or *Detail – Transpose* results views. The following figure shows an example of a cross-probed point from a histogram to the corresponding design point in the *Detail* view.



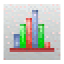
If you do not need to cross-probe results from histogram, clear the *Show Points* check box on the Plot Histograms form or on the ADE XL Printing/Plotting Options form. You can also set the showHistogramPoints environment variable to `nil`. This fills the bars with solid red color and disables cross-selection from histogram points. Alternatively, you can clear the *Symbols On* command on the context-sensitive menu of a histogram.

You can also change the style of data points by using the *Symbols* command on the context-sensitive menu of a histogram and select any shape to be used as symbol.

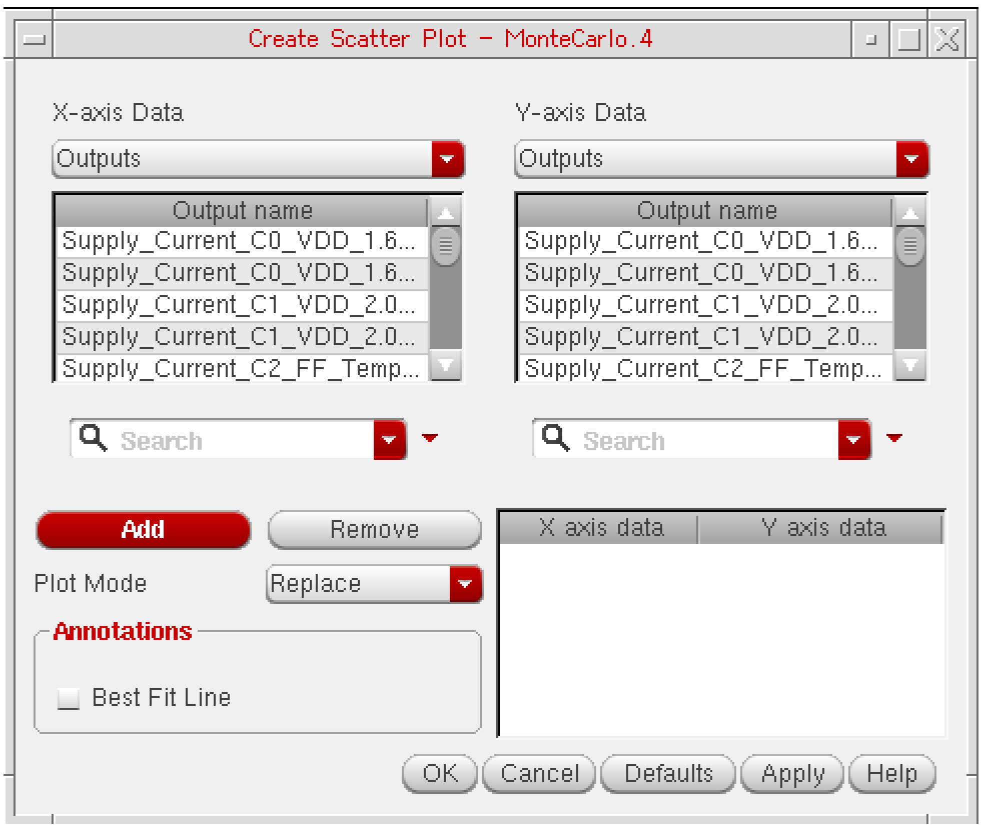


Plotting Scatter Plots

To plot a scatter plot depicting the relationship between the pairs of outputs or parameters in the Monte Carlo results, do one of the following:

1. On the Results tab of the Outputs pane, switch to the *Yield* view.
2. Click the  button on the toolbar, select a test, and choose *Scatter Plot.*

   The Create Scatter Plot form is displayed.



   Note that the names of all the outputs are displayed in the *X-axis Data* and *Y-axis Data* lists. To view the list of parameters, select `Statistical Parameters` from the drop-down lists given on top.

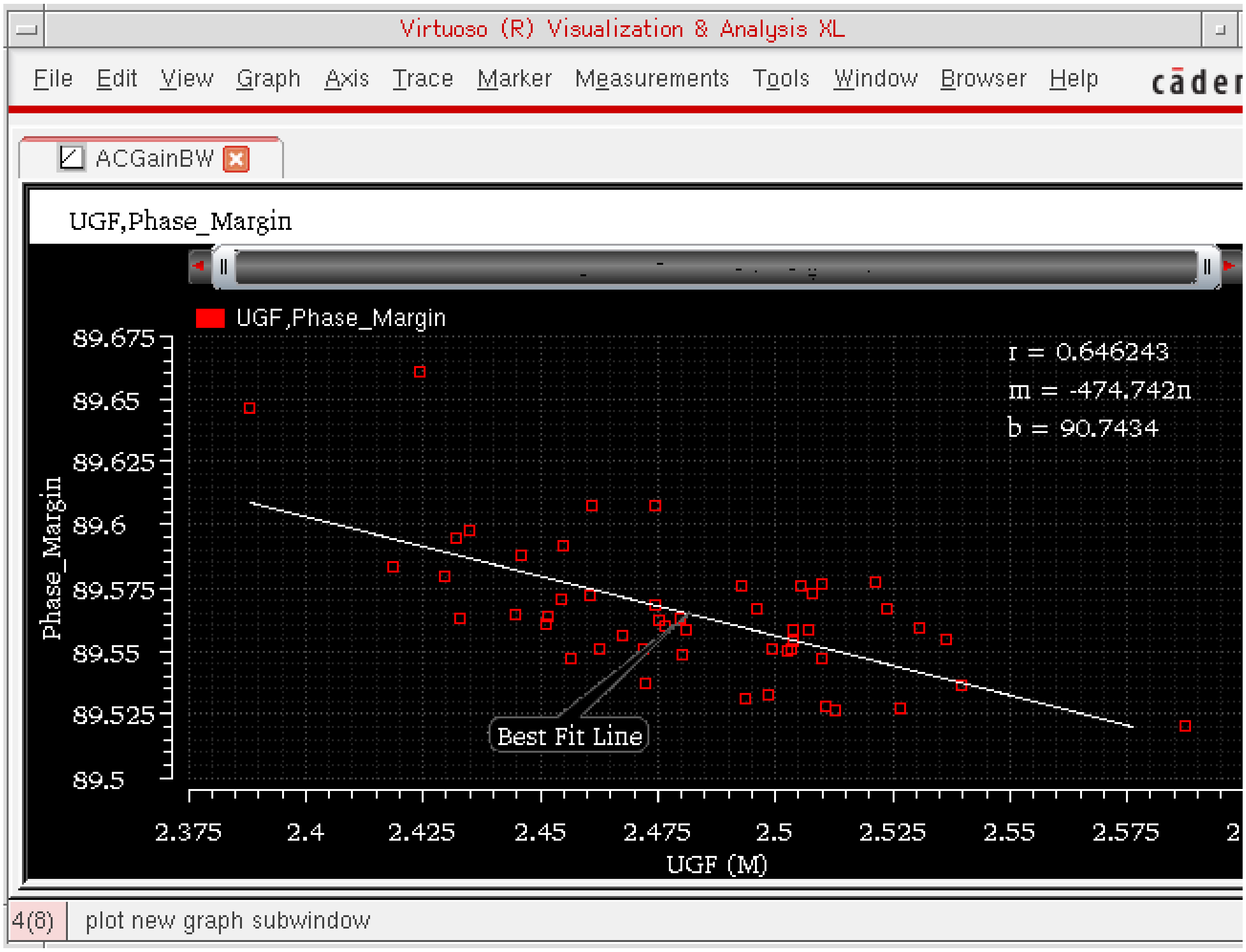
3. To make an output pair, select an output from each one of the *X-axis Data* and *Y-axis Data* list.
4. Click *Add*.

   Similarly, you can select more outputs or parameters. The selected pairs appears in the list box shown in the bottom right corner of the form.

5. (Optional) Select a plotting mode from the *Plot Mode* list. The default value is `Replace`. For more details on the plotting modes, refer to <u>Selecting the Plot Mode</u>.
6. (Optional) Click *Best Fit Line* to draw a least-squares-fit line for each parameter pair.
7. Click *OK*.

   The scatter plot is displayed for the selected output or parameter pairs in the Virtuoso Visualization and Analysis XL window.

   An example is shown below.



Note the following values in the trace legend:

- `r` shows the correlation coefficient, that is, how correlated the X- and Y-axis data are to each other. The value of `r` ranges between `-1.0` and `1.0`.

- `b` is the constant value and `m` is the value of slope in the linear equation describing the best fit line of the plotted data. The linear equation that is used to find the best fit line is `y = b + mx`.

Viewing Sensitivity Results

To view data on the sensitivity of variables and measurement expressions for a Monte Carlo run, do one the following:

- Click the *Sensitivity Results* button on the Results tab.
- In the History tab on the Data View pane, right-click on the history item for a Sensitivity Analysis or Monte Carlo run and choose *Sensitivity Results*.

  The Sensitivity Analysis form appears displaying the sensitivity results.

  For more information about the Sensitivity Analysis form, see the *Virtuoso Analog Design Environment GXL User Guide*.

Viewing Statistical Parameters for Monte Carlo Samples

After running a Monte Carlo analysis, you can switch to the *Detail* view on the Results tab of the Outputs pane to view the detailed results for each Monte Carlo sample. If the results for any sample need further investigation, you can view the statistical parameters for that sample so that you can debug your design.

To view and print the statistical parameters for a Monte Carlo sample, do the following:

1. In the *Select the results view* drop-down list on the Results tab of the Outputs pane, select *Detail*.

   The Detail view is displayed.

2. Right-click on a result for a sample in the *Nominal* column and select *Print Statistical Parameters*.

   The statistical parameters for that sample are displayed in a Results Display Window.

   For information about printing the contents in the Results Display Window, see Printing Results from the Results Display Window on page 683.

Running Multi-Technology Simulations for Monte Carlo Analysis

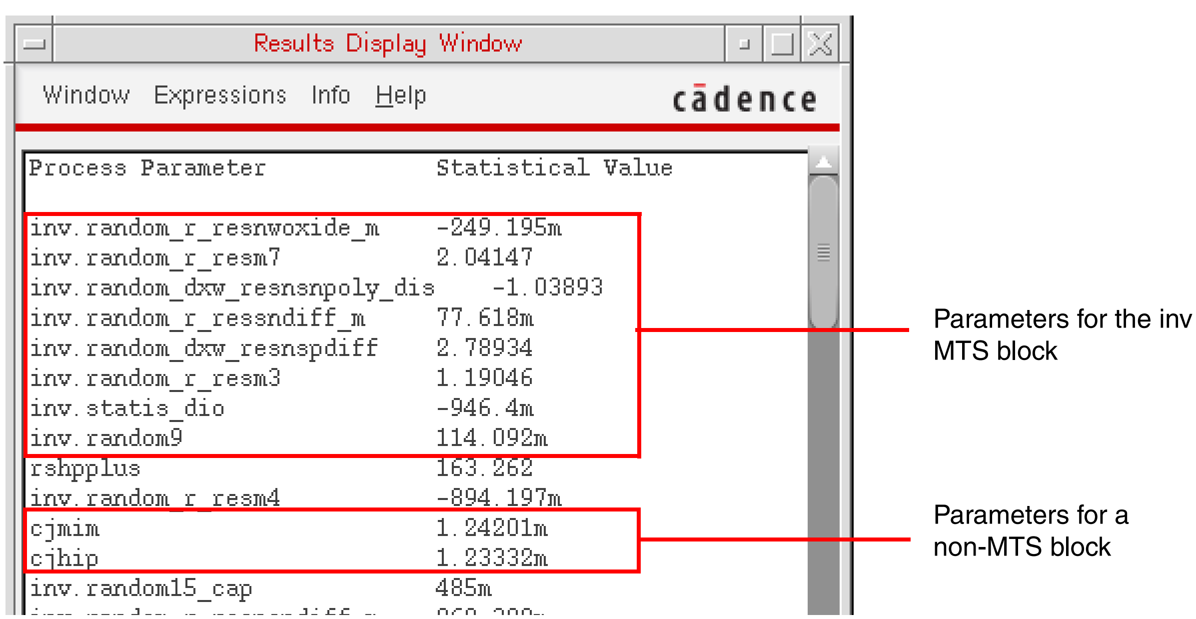
You can run a Monte Carlo analysis in the multi-technology mode by enabling the Multi-Technology Mode option for a simulator and specifying the related Multi-Technology Simulation (MTS) options. For details on how to enable these settings, refer to the Multi-Technology Simulation chapter in the *Analog Design Environment GXL user guide*.

After identifying the MTS blocks and specifying MTS options for them, run the Monte Carlo analysis. If you have specified different model libraries for the MTS blocks, the statistical and process parameters used for those blocks are different than the parameters used for the non-MTS blocks.

Viewing Results of Monte Carlo Analysis with MTS

After a multi-technology simulation is run, you can view the results on the *Results* tab. To view the process and statistical parameters used for simulation of a data point, change the results view to the *Yield* view, right-click on a result data cell and choose *Print Statistical Parameters*.

The Results Display Window shows a list of all the process and statistical parameters along with the statistical value used for simulation. Note that the names of parameters for the MTS blocks are prefixed with the name of their corresponding block. For example, in the following figure, the names of parameters for the `inv` MTS block are prefixed with `inv`.



Similarly, in the Sensitivity Analysis results window, the parameter names for MTS blocks are prefixed with the name of their corresponding block. For more details, refer to Support for MTS in Sensitivity Analysis in Analog Design Environment GXL user guide.

8

Performing Reliability Analysis

ADE XL supports Cadence® Virtuoso® RelXpert Reliability Simulator and native reliability analysis in Spectre and APS. The reliability analysis analyzes the effect of time on circuit performance drift and predicts the reliability of designs in terms of performance. In ADE XL, you can run the reliability simulation for fresh test (when time is zero), stress test (to generate degradation data), and aged test (at specific intervals, such as 1 yr, 3 yrs, or 10 yrs). In the stress test, extreme environmental conditions are used to stress devices before aging analysis.

Note: The RelXpert Reliability Simulator interface in ADE XL is supported with the Spectre Simulator in MMSIM 7.2 or a later version. For more information, see *Virtuoso RelXpert Reliability Simulator User Guide*.

This chapter describes how to set up and run reliability simulation in ADE XL.

See the following topics for more information:

Important

Currently, reliability analysis is run with Sweeps and Corners run mode only. You can sweep design variables including age. This analysis is not compatible with any other run mode.

Simulator Modes for Reliability Analysis

You can run reliability analysis in two simulator modes:

- Spectre native: The Spectre native mode runs simulations using the Spectre simulator. The is the default simulator mode and gives higher performance for large circuits. Starting from MMSIM10.1.1 ISR6, Spectre native reliability analysis supports the BSIM3V3, BSIM4, BSIMSOI and PSP102 models.

  Note: To know more about the Spectre native simulator mode and the models supported by it, refer to the Reliability Simulation Block section in *Virtuoso® Spectre® Circuit Simulator and Accelerated Parallel Simulator User Guide*.

- RelXpert: The RelXpert mode runs simulations using the RelXpert simulator. This mode accepts agemos parameters for BSIM3v3 and BSIM4 and supports the agemos method for aged `.model` card generation.

  You can choose a simulator mode by using the Reliability form.

Specifying the Reliability Analysis Setup

Before you specify the reliability analysis setup, ensure that you do the following:

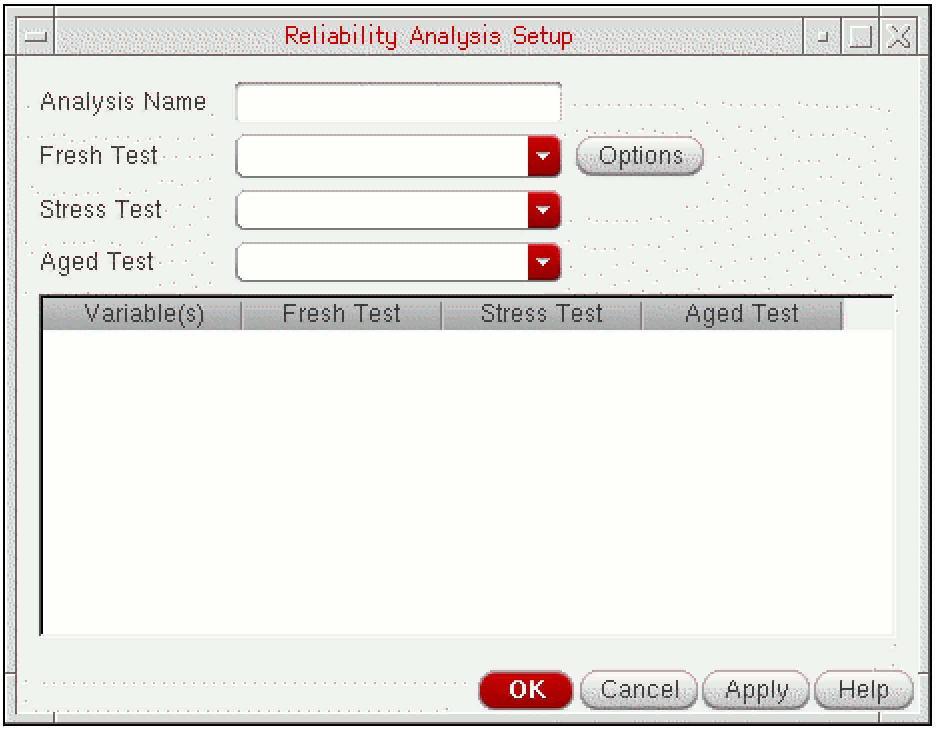
- In the Data View assistant pane, select the tests on which you want to run the reliability simulation (fresh, stress, and aged). This is required because only the tests you select in the Data View assistant pane are visible in the setup form.

- On the *Run* toolbar, in the *Select a Run Mode* drop-down list, select *Single Run, Sweeps and Corners*. Reliability analysis is supported only for this run mode.

To specify the reliability analysis setup, follow the steps listed below:

1. In the Data View assistant pane, expand the *Reliability Analyses* node and click where it says *Click to add reliability analysis setup*.

The Reliability Analysis Setup form appears.



2. In the *Analysis Name* field, specify a unique name for the reliability analysis setup.
3. From the *Fresh Test* drop-down list, select the name of the test you want to use for running fresh simulation.
4. From the *Stress Test* drop-down list, select the name of the test you want to use for running stress simulation.
5. From the *Aged Test* drop-down list, select the name of the test you want to use for running an aging simulation.

   Note the following important points with respect to the selection of tests for the fresh, stress, and aged test:

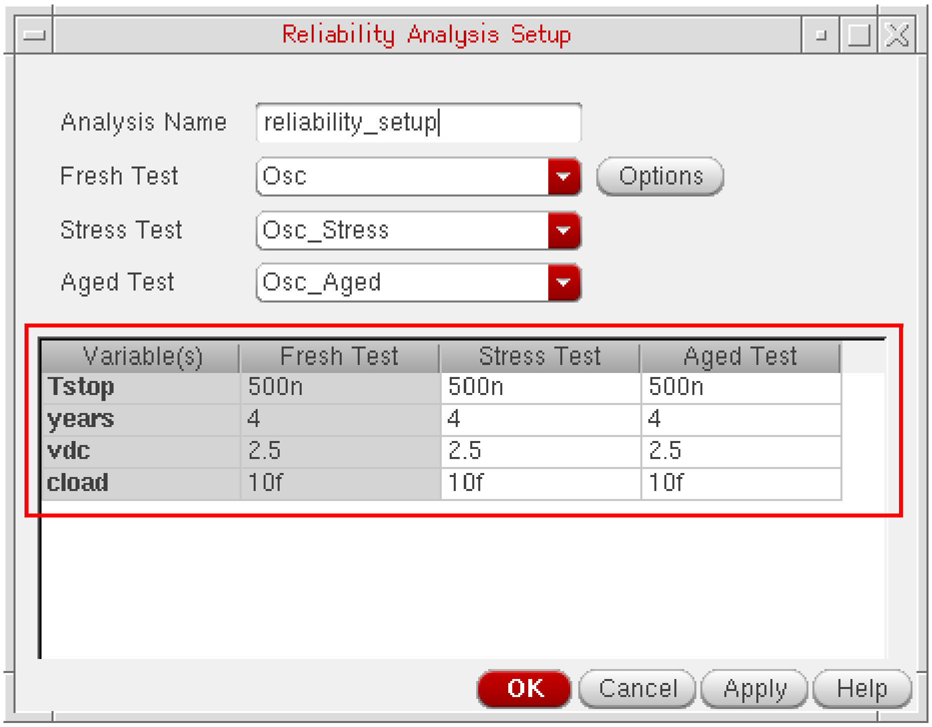
   ❑ It is possible to specify different test names for fresh, stress, and aged tests. However, the reliability options, set in step 7, are used only from the fresh test.

   ❑ Simulation for an aged test uses the simulation data generated by the stress test. Therefore, a stress simulation must be run before an aging simulation is run.

   ❑ If you select a test that is already specified in another reliability analysis setup, ADE XL displays a warning stating that any changes that you make to this test take effect in the other reliability analysis setups too.

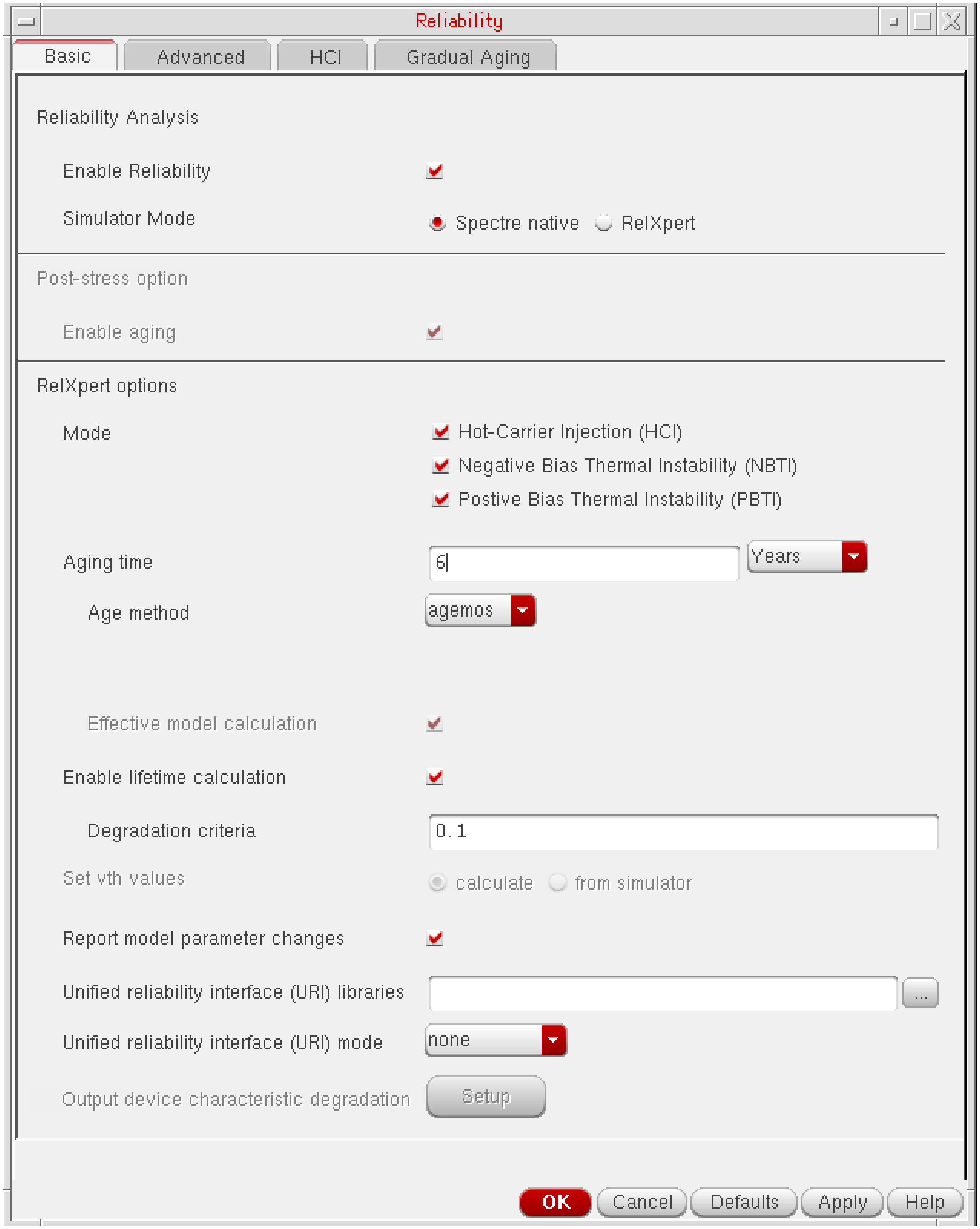
The table displayed at the bottom of the form shows the values of variables to be used for fresh, stress, and aged tests,

6. If required, for the stress and aging simulations, you can modify values of the variables in this table. For the fresh test, you can modify the values in the ADE XL Test Editor.



7. Click *Options*.

   The Reliability form appears.


Reliability
Basic
Advanced
HCI
Gradual Aging
Reliability Analysis
Enable Reliability
Simulator Mode
Spectre native
RelXpert
Post-stress option
Enable aging
RelXpert options
Mode
Hot-Carrier Injection (HCI)
Negative Bias Thermal Instability (NBTI)
Postive Bias Thermal Instability (PBTI)
Aging time
6
Years
Age method
agemos
Effective model calculation
Enable lifetime calculation
Degradation criteria
0.1
Set vth values
calculate
from simulator
Report model parameter changes
Unified reliability interface (URI) libraries
...
Unified reliability interface (URI) mode
none
Output device characteristic degradation
Setup
OK
Cancel
Defaults
Apply
Help

8. In the Reliability form, specify options for the simulation.

These options for reliability analysis are stored in the setup for the fresh test, but the same options are applied to all fresh, stress, and aging simulations.

Note: If you have already specified reliability options for the fresh test in the ADE L environment or ADE XL Test Editor, the same settings are displayed in the Reliability form. If required, you can modify the settings in the ADE XL environment.

For details of the Reliability form, refer to Reliability Form.

9. After specifying the reliability options, close the Reliability form.

10. Click *OK* to close the Reliability Analysis Setup form.

The reliability setup appears under the *Reliability Analysis* tree on the Data View assistant pane.

If required, you can add notes to the reliability analysis setup you have created. For this, right-click on the reliability analysis setup name on the Data View assistant pane and click *Notes* to open the Add/Edit Notes form. Add notes in the *Notes* field and click *OK*. These notes are displayed in the tooltip for the setup and saved in the setup database.

For related information, see Adding Notes to a Test.

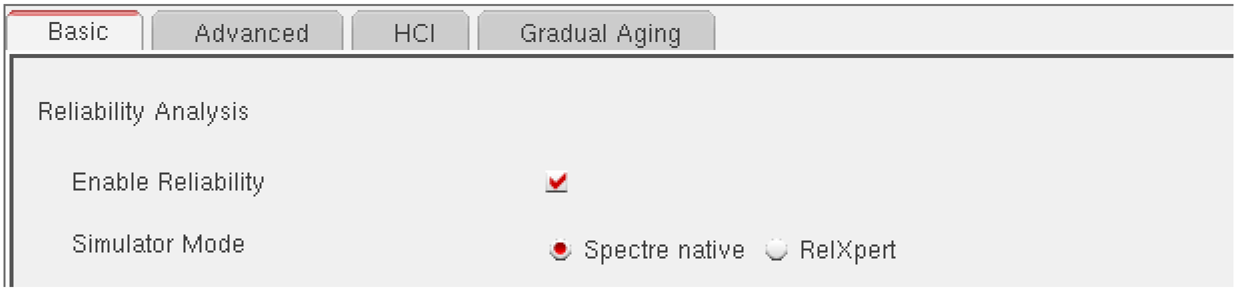
Reliability Form

The Reliability form specifies options for reliability analysis. This form contains four tabs:

Basic

This tab contains basic options to be used for reliability analysis. In the *Reliability Analysis* section, you can enable/disable the reliability analysis for the test.

An important option to set on this tab is to choose a simulator mode that you want to use for reliability analysis.



For details about the two simulator modes, refer to Simulator Modes for Reliability Analysis.

Advanced

This tab contains advanced options to be used for reliability analysis.

For more details about the options on this tab, refer to Reliability in *Analog Design Environment L User Guide*.

HCI

By using the options on this tab, you can control of the accuracy and approach in calculating the HCI effects during the aging analysis.

For more details about the options on this tab, refer to Reliability in *Analog Design Environment L User Guide*.

Gradual Aging

By using the options on this tab, you can set options to measure the effects of gradual aging by sweeping age. For this, you need to enable *Gradual aging* and specify an aging mode and related options.

For more details about the options on this tab, refer to Reliability in *Analog Design Environment L User Guide*.

Important

> There are many options on the Reliability form that are not enabled for the Spectre native mode. For the complete list of those options, refer to Reliability Options Not Supported In Spectre Native Simulator Mode in *Analog Design Environment L User Guide*.

Running the Reliability Simulation

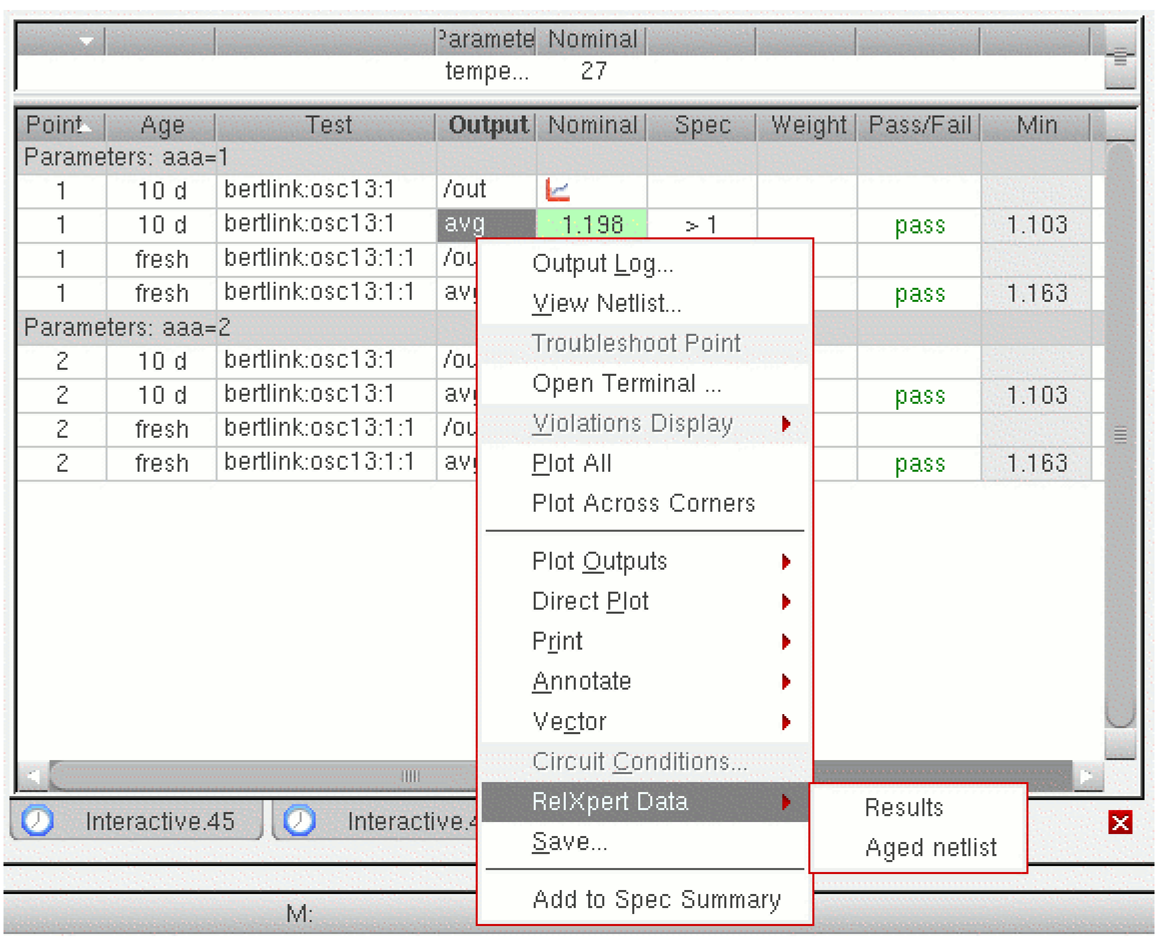
Before you run reliability simulation, ensure that the following prerequisites are met:

- A Spectre or APS transient analysis is set up and enabled for the fresh, stress, and aged tests. You can set these by using the ADE XL Test Editor.
- The *Enable Reliability* check box is selected in the RelXpert Options form.

To run the reliability simulation:

➡ Click the *Run Simulation* button  .

The fresh, stress, and aging simulations are run for the specified tests. To view the RelXpert simulation result data, right-click on the *Results* tab and choose an appropriate option from the menu shown in the following figure.



For more details on how to view RelXpert simulation result data, refer to Working with RelXpert Data.

Important Points to Note

- You can also run multiple RelXpert setups simultaneously. When you run multiple setups, an additional column, *Relx*, is visible before the *Age* Column on the *Results* tab. The *Relx* column specifies the name of the corresponding RelXpert setup.

  The number of results sets generated from the reliability analysis is determined by the total number of data points on which the simulations specified in the reliability analysis setup is run. Consider a reliability analysis with the following data that needs to be run for fresh and aged tests (two age points):

  - 1 variable with 2 sweep points
  - 3 corners (2 corners + 1 nominal corner)

  In this example, each simulation (one fresh and two aging) is run 6 times. Therefore, the total number of data points for which the reliability analysis is run is 18.

  For more information, see the Working with RelXpert Data.

- If we have enabled Reliability Analysis in ADE XL, simulations are run only for those tests that are part of any of the Reliability Analysis setup. Therefore, if there are any tests that are enabled in ADE XL user interface, but are not used as fresh, stress, or aged tests in the Reliability Analysis setup, those tests are ignored. In such a case, the following warning message is displayed in CIW.

```
WARNING (ADEXL-7005): The following tests are not being used in any of the
enabled Reliability Analysis setups and hence it will be disabled for the
current run
```

- RelXpert Reliability Simulator always runs in batch mode. Therefore, ADE XL too automatically runs in batch mode when you enable RelXpert. The following message is displayed in CIW indicating the change in mode:

```
WARNING: The Spectre run mode needs to be 'batch' when running RelXpert.
Automatically setting the run mode to 'batch'.
```

- Consider the case of an ADE XL setup in which reliability analysis is not set up. If you modify a testbench in the ADE XL Test Editor, and enable reliability analysis, after you close the Test Editor, a default reliability setup named `Default_Relx` is created in the Data View pane. However, it is important to note that the default ADE XL reliability setup will be created only for the test for which reliability setup is enabled in the Test Editor. Once a default setup has been created for a test, if you create or modify another test in the Test Editor, no default setup will be created for that test.

Working with RelXpert Data

With RelXpert data, you can do the following:

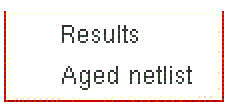
- Viewing Simulation Results
- Viewing Aged Netlist
- Plotting Results
- Plotting Results Across Corners
- Printing or Plotting Stress Results for Gradual Aging Run
- Annotating Simulation Results to Schematic View

Viewing Simulation Results

To view simulation results, do the following:

1. On the *Results* tab, select the required test.
2. Right-click the selected test and choose *Reliability Data*.

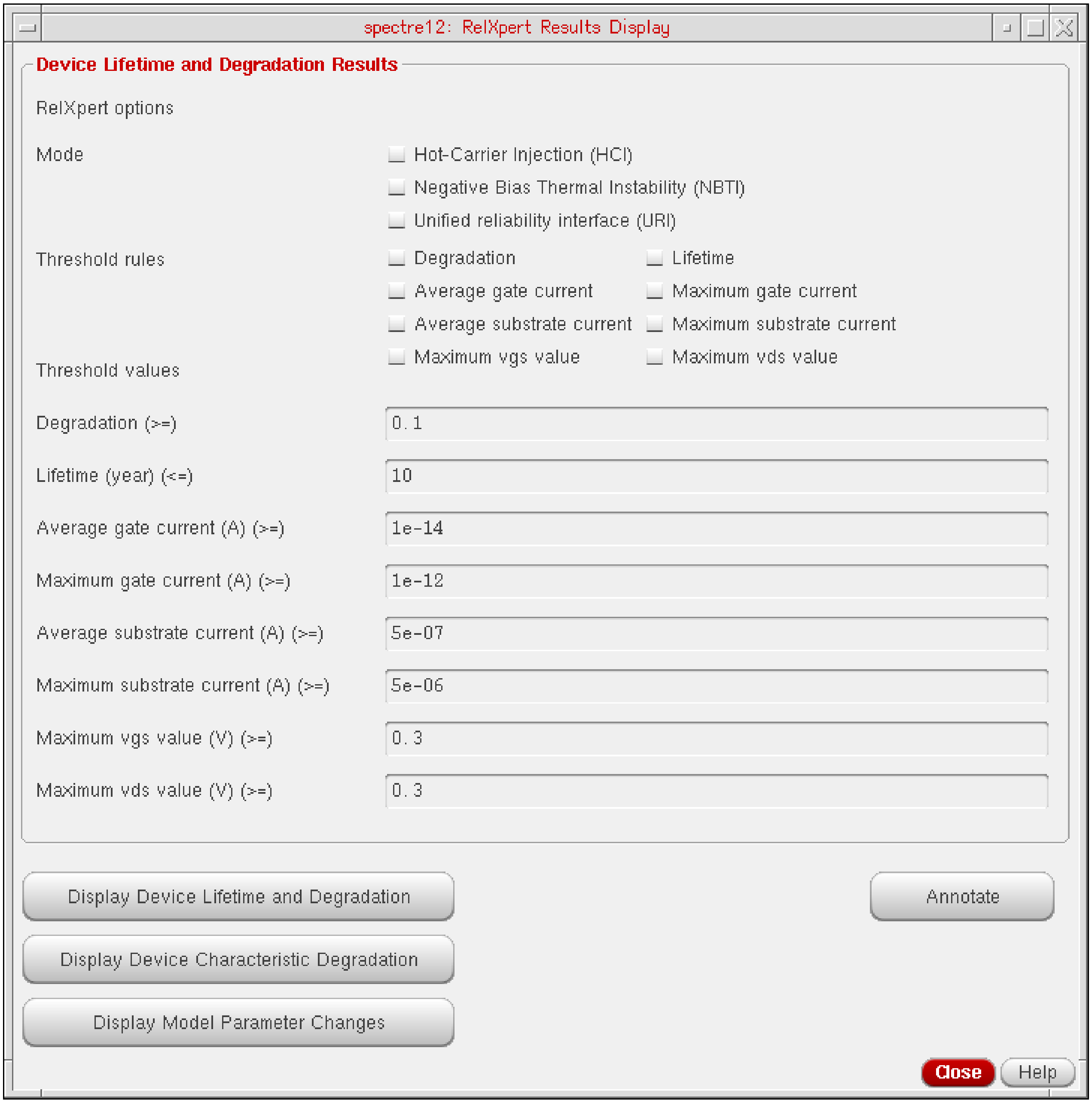
   If you have used MMSIM 11.1 to run simulation, the following submenu is displayed.

   Note: Depending on the type of test: fresh, stress or aged; the simulator mode: native or RelXpert); and the reliability setup, only those commands are available in this submenu for which the corresponding results are available.

3. Choose *Results*.

The RelXpert Results Display form appears.



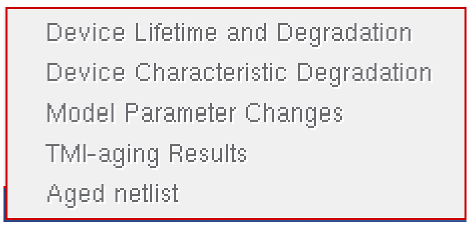
4. In the RelXpert Results Display form, you can do the following:

`Click...`	To...
`Device Lifetime and Degradation`	Display the lifetime and degradation values for all degraded devices.

`Device Characteristic Degradation`	Display the device degradation for the specified aging time period.
`Model Parameter Changes`	Display fresh and aged parameter information.

For more details on this forms to view reliability results, refer to Virtuoso Analog Design Environment L User Guide.

Note: If you have used MMSIM 12.1 to run simulation, the following submenu is displayed and you can directly use these commands to view the reliability data.



Viewing Aged Netlist

To view the aged netlist for a test:

1. In the Virtuoso ADE XL window on the *Results* tab, select the result of an aged test.
2. Right-click the selected test and choose *Reliability Data – Aged netlist*.

The aged netlist generated by the RelXpert Reliability Simulator is displayed in a new window. A sample is shown below.

Note: The *Aged Netlist* command is available only when aging results are generated from an aging simulation run. This command is disabled if an aging simulation is not run as part of the reliability simulation.

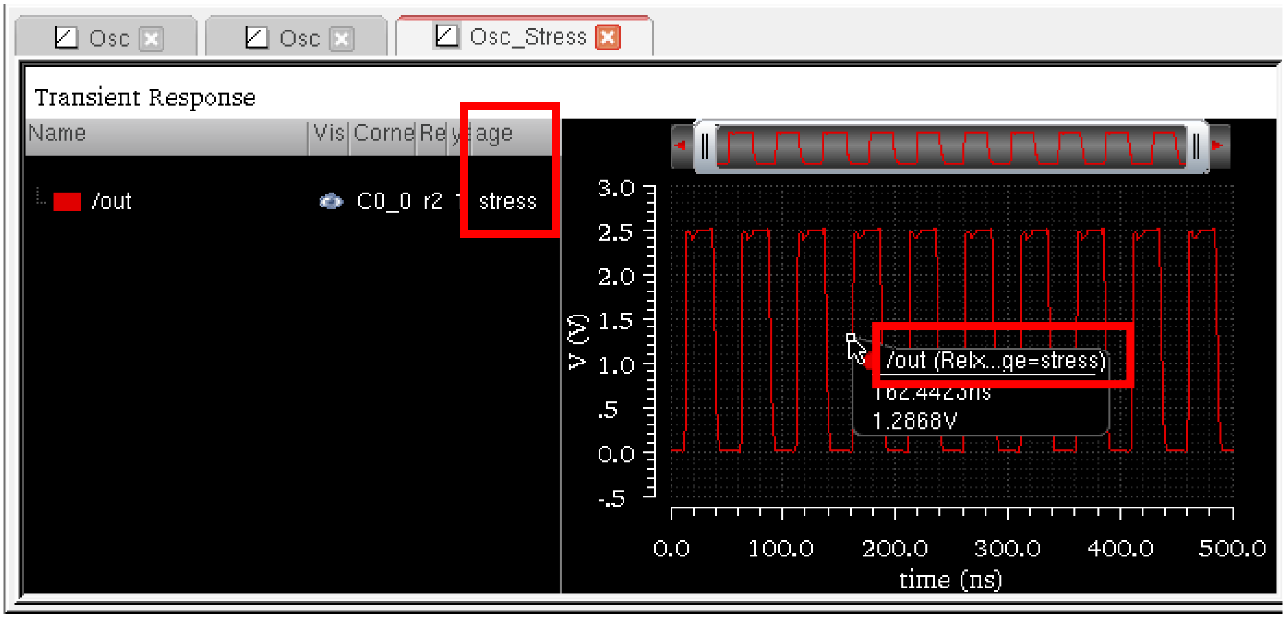
Plotting Results

You can plot the fresh, stress, or aged results in the Virtuoso Visualization and Analysis XL window.

To plot the results:

➡ On the *Results* tab, right-click on the fresh, stress, or aged result and choose *Plot*.

The results are plotted in the Virtuoso Visualization and Analysis XL window. Note that in the plot of fresh test, the graph legend shows age value as `0` and for the age test, the legend shows the age in years. However, for the stress result, the trace legend and trace name for the stress plot shows `age=stress` to indicate that the results are plotted for stress results, as shown below.



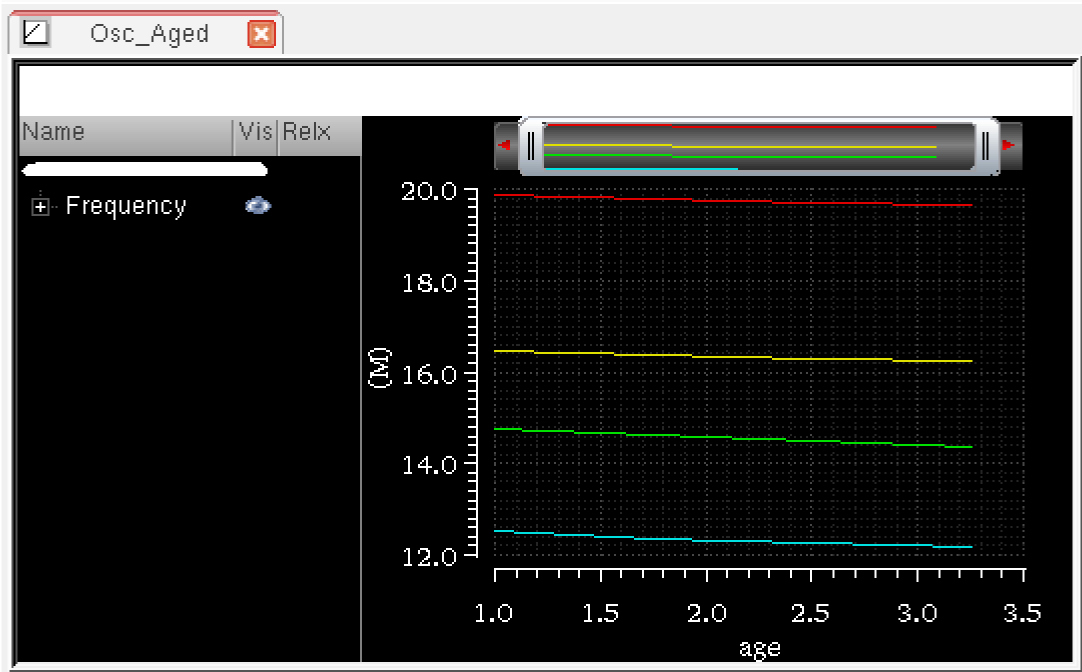
Plotting Results Across Corners

You can plot the fresh and aged results across corners.

To plot results across corners:

➡ On the *Results* tab, right-click on the fresh or aged result and choose *Plot Across Corners*.

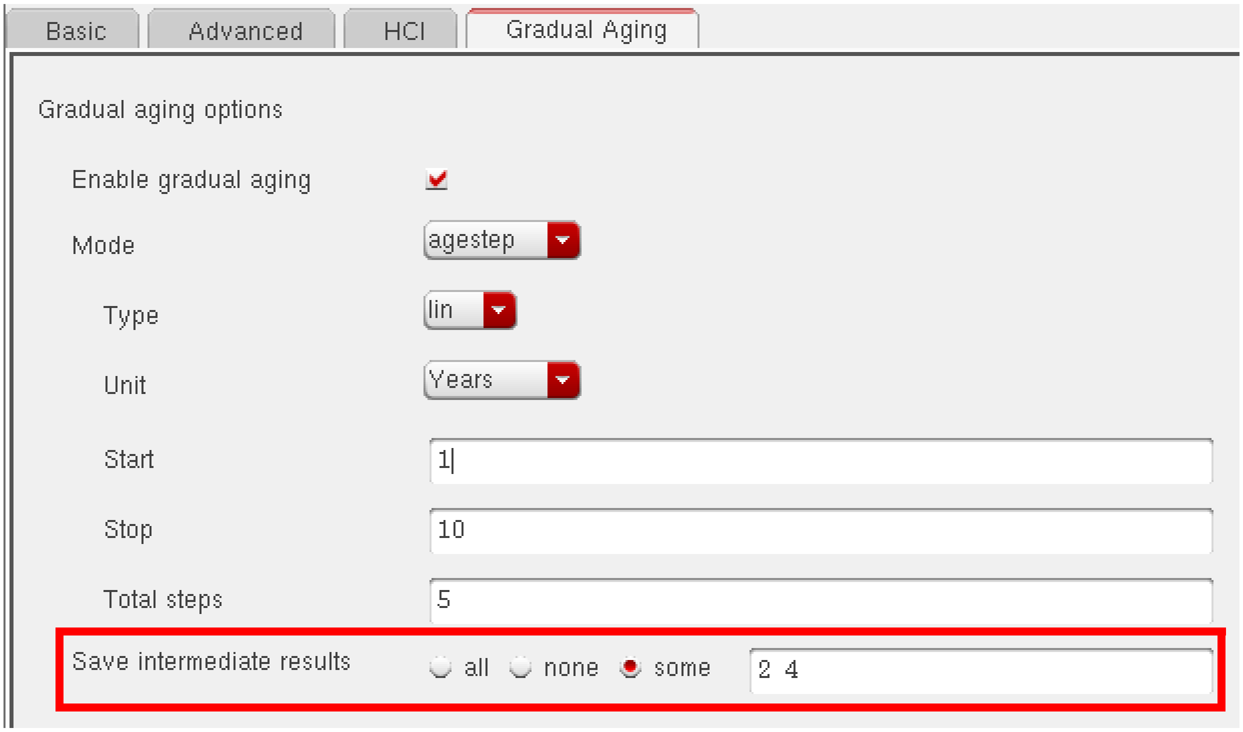
The results are plotted in the Virtuoso Visualization and Analysis XL window, as shown below.



Note: When you plot the results of Reliability analysis across corners, age is plotted on the x-axis.

Printing or Plotting Stress Results for Gradual Aging Run

If you had chosen to save the results data for gradual aging, you can print or plot the stress results for every age point for which data was saved.



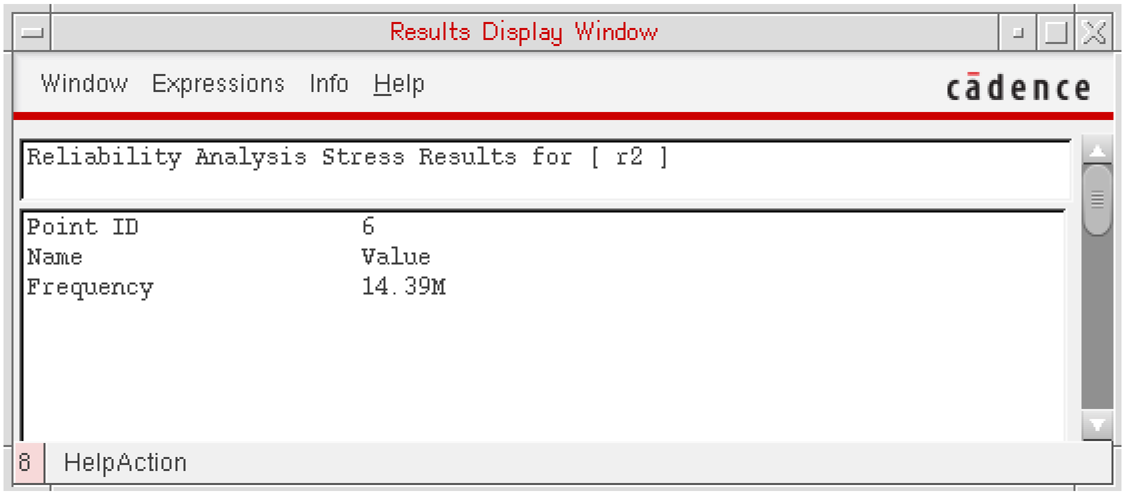
- Printing Stress Results for an Age Point
- Plotting Stress Results for an Age Point

Printing Stress Results for an Age Point

To print stress results for an age point:

1. On the *Results* tab, select the result of that age point.
2. Right-click and choose *Print Stress Results*.

   A text file containing the stress results is displayed, as shown below.



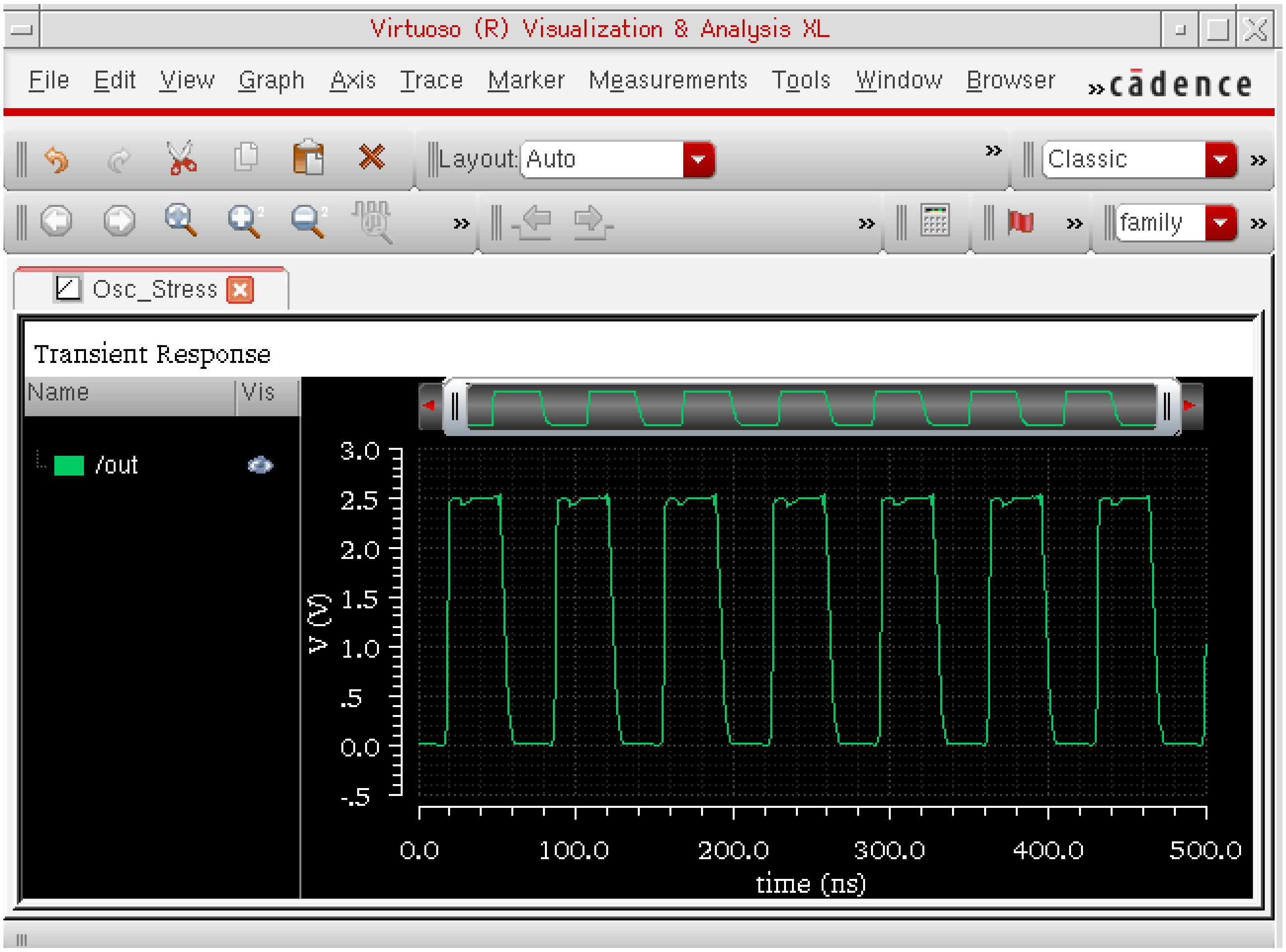
Note: The *Print Stress Results* command is available only for the aging points of the gradual aging run. It is not available for fresh and stress points or age point of non-gradual aging run.

Plotting Stress Results for an Age Point

To plot stress results for an age point:

1. On the *Results* tab, select the result of that age point.
2. Right-click and choose *Plot Stress Results*.

   The results are plotted in the Virtuoso Visualization and Analysis XL window, as shown below.



Note: The *Plot Stress Results* command is available only for the aging points of the gradual aging run. It is not available for fresh and stress points or age point of non-gradual aging run.

Annotating Simulation Results to Schematic View

To annotate the device lifetime and degradation results to schematic view:

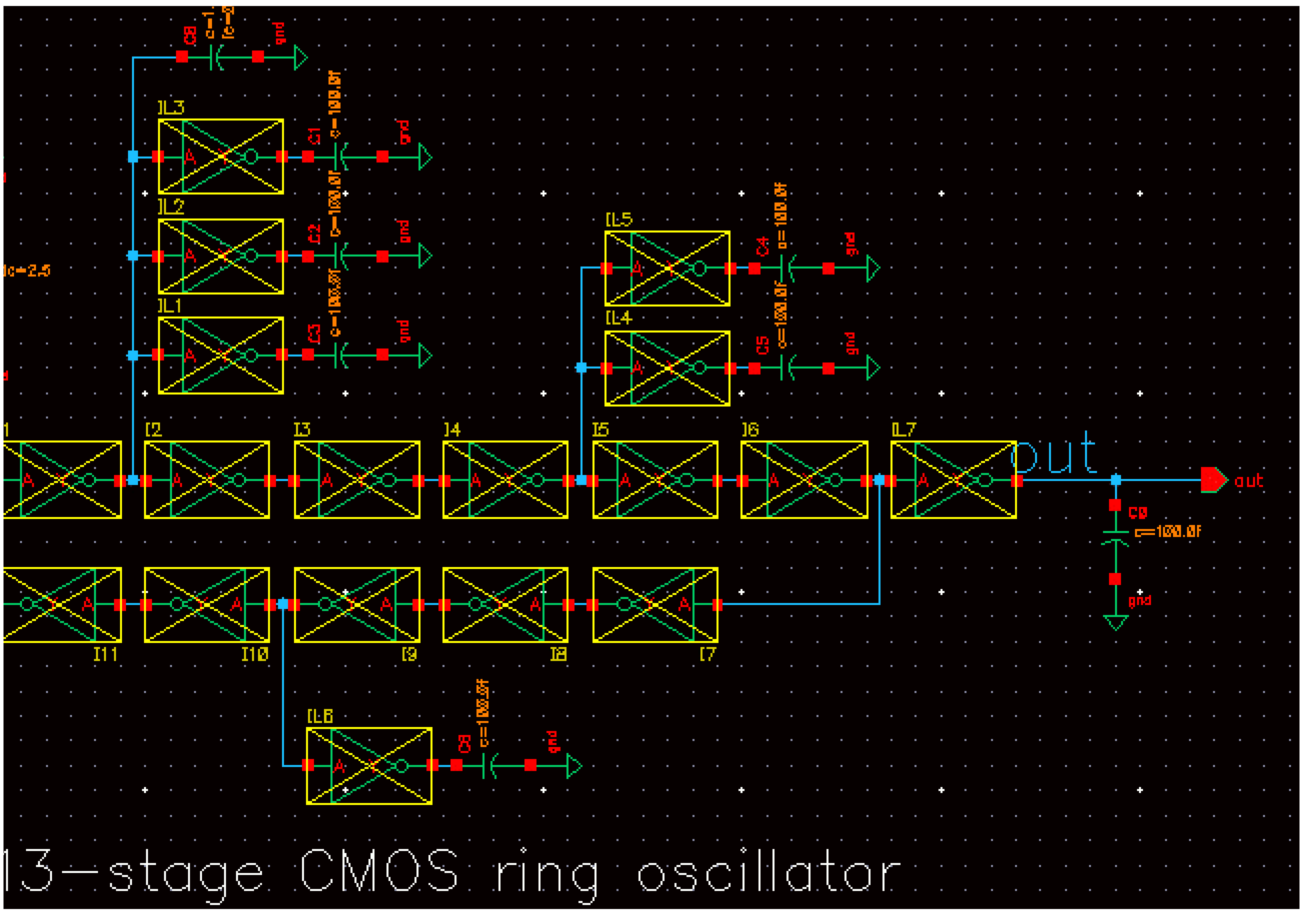
1. In the Virtuoso ADE XL window, on the *Results* tab, select the required test.
2. Right-click the selected test and choose *Reliability Data – Results*.

   The RelXpert Results Display form appears.
3. Select the RelXpert modes for which you want to backannotate the results.

Results are backannotated based on the specified threshold values. For example, if the Lifetime (year) (<=) field specifies a threshold value of 10, results are backannotated only to those instances whose lifetime is less than or equal to 10 years.

4. Select the threshold rules based on which you want to backannotate the results. The threshold rules corresponding to the selected modes are displayed in the form. The fields containing the default threshold values for the selected rules become available.

5. Click the *Annotate* button. The instances to which the results are backannotated are highlighted in yellow in the schematic.



Note: In a hierarchical design, a block is highlighted if results are backannotated to an instance in that block. When you descend into the block, the instances to which the results are backannotated are also highlighted.

9

Working with Model Files and CDF

Model files contain definitions for models used in your design that are not defined in the library. You can specify the model files you want to reference for simulation on the Model Library Setup form. You can associate a model name (from one of your referenced model files) with a component on your schematic by setting the *Model name* CDF parameter. The *Model name* could refer either to a model definition or to a subcircuit (also called *macro*) definition. To netlist a subcircuit correctly, you must have a symbol cellview, a stopping cellview, and appropriate CDF information on the cell.

For example, the `opamp` schematic in the `aExamples` library contains instances of `npn` and `pnp` bipolar transistors (the *Model name* is set to `npn` or `pnp` accordingly). Cadence provides model definitions for Spectre circuit simulation of these parts in the file

your_install_dir`/tools/dfII/samples/artist/models/spectre/bipolar.scs`

You can declare this model library file by typing the path and file name above in the *Model Library File* field on the Model Library Setup form.

Note: See “Simple Device Models” in the Direct Simulation Models chapter of the *Direct Simulation Modeling Guide* for more information.

See also

- Specifying Model Libraries on page 96
- Associating a Model or Subcircuit Name with an Instance on page 352
- Editing Component CDF on page 353
- Adding a Model Name Parameter to a Component’s CDF on page 355
- Using Component CDF to Specify Simulation Information on page 357
- Creating a Stopping Cellview on page 357
- Varying the Model File and Section during Simulation on page 359
- Editing a Model File on page 362
- Disabling a Model File on page 362

Associating a Model or Subcircuit Name with an Instance

To associate a model or subcircuit name with an instance on your schematic do the following:

1. On the schematic, select the component.

   The instance name and its parameters appear in the scrolling list area on the top part of the Parameters tab on the Variables and Parameters assistant pane.

2. Locate the model-name parameter (such as *model*).

   You can use a filter to narrow the device instance parameter selection list.

   If your component does not have a model name parameter in its CDF, see "Adding a Model Name Parameter to a Component's CDF" on page 355.

   If your component does have a model name parameter but the parameter value is not editable, see "Making the Model Name Parameter Editable" on page 356.

   See also "Toggling the View on the Variables tab of the Variables and Parameters Assistant Pane" on page 209.

3. Double-click in the *Value* column and type a model name.

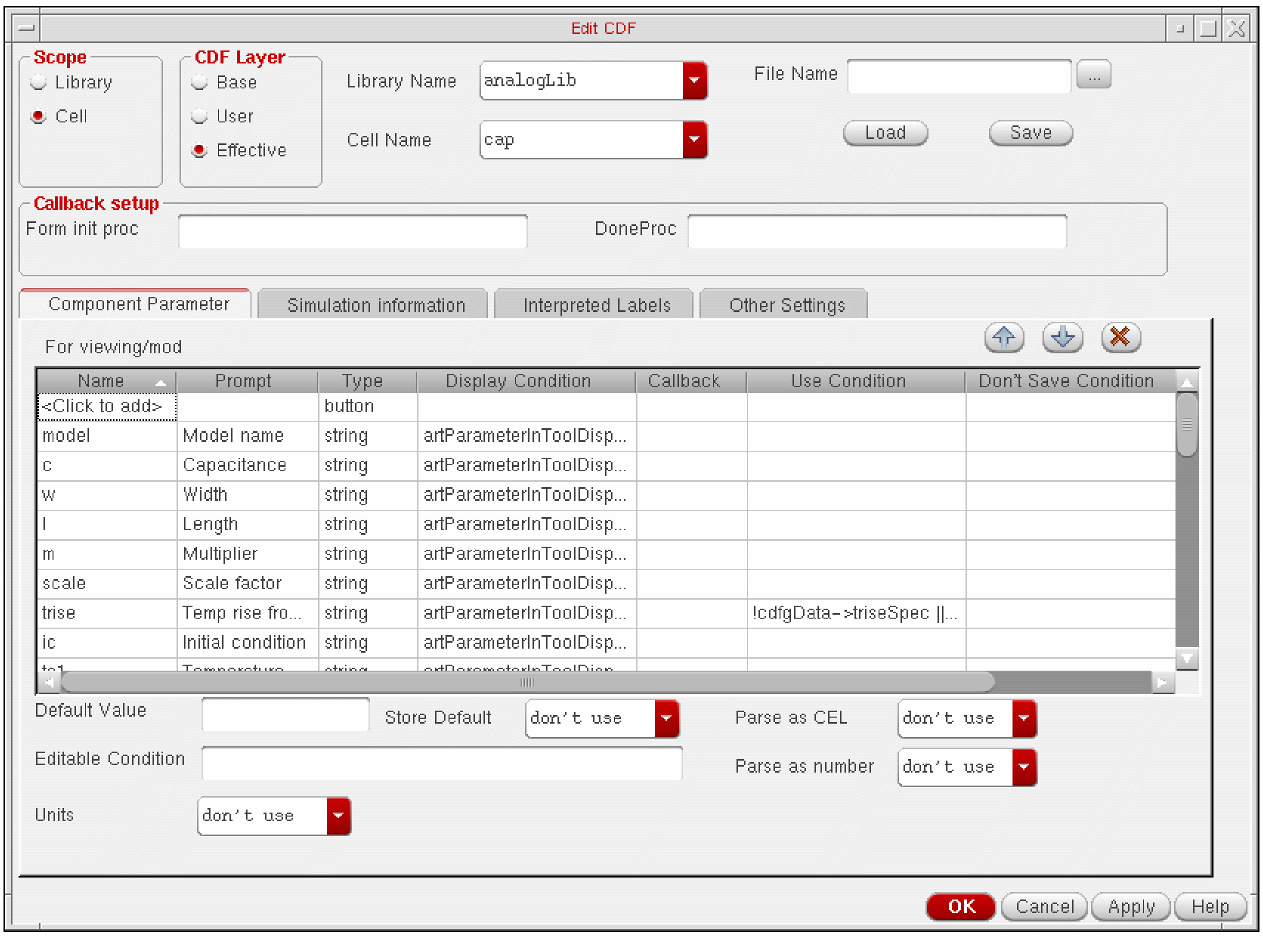
4. Use the Model Library Setup form to specify the file containing the model or subcircuit definition whose name you typed in the *Value* column (see "Specifying Model Libraries" on page 96).

Editing Component CDF

To Edit CDF, do the following:

1. In the CIW, choose *Tools – CDF – Edit*.

   The Edit CDF form appears.



2. For *Scope*, select the *Cell* radio button.

   See “CDF Selection” in the "CDF Commands" chapter of the *Component Description Format User Guide* for information about this selection.

3. In the *CDF Layer* group box, select a CDF type: *Effective*, *Base*, or *User*.

   See “CDF Layer” in the "CDF Commands" chapter of the *Component Description Format User Guide* for information about this selection.

4. Click *Browse*.

The Library Browser form appears.

5. In the *Library Name* drop-down list, select a library.
6. In the *Cell* drop-down list, select a cell.

   The CDF information for the cell is displayed on the Edit CDF form.

See also

- Adding a Model Name Parameter to a Component's CDF on page 355
- Making the Model Name Parameter Editable on page 356
- Using Component CDF to Specify Simulation Information on page 357

Adding a Model Name Parameter to a Component's CDF

To add a parameter for model name to your component, do the following:

1. Open the Edit CDF form.
2. Specify the cell (component) whose CDF you want to edit.
3. In the *Name* column on the *Component Parameter* tab, click where it says *<Click to add>.*

   Specify your model name parameter as follows:

Column	Value
Name	Type `model`.
Prompt	Type `Model name`.
Type	Select *string* as the parameter type.
Parse as CEL	Select *no* from the drop-down list. You do not want your model name parsed as an expression.
P arse as number	Select *no* from the drop-down list. You do not want your model name parsed as a number.

4. Click *Apply* to apply your changes and leave the form open, or *OK* to apply your changes and close the form.

For more information, see "Defining Parameters" in the *Component Description Format User Guide*.

Making the Model Name Parameter Editable

To make the model name parameter editable (so that you can vary it during simulation), do the following:

1. Use the Edit CDF form to edit the component's CDF.
2. In the *CDF Layer* group box, select *Base*.
3. In the *Component Parameter* tab, select the model-name parameter (such as *model*).
4. In the *Editable Condition* field, type a non-nil value.
5. Click *OK*.
6. On the Edit CDF form, click *OK*.

   When you select this component instance on your schematic, the model-name parameter appears in the list of *CDF Editable* device instance parameters in the Parameters tab on the Variables and Parameters assistant pane.

Using Component CDF to Specify Simulation Information

To specify how you want the software to netlist your component, what parameters to pass to the underlying subcircuit or model definition, and the order of the input terminals (for netlisting), do the following:

1. Use the Edit CDF form to edit the component's CDF.
2. Click the *Simulation Information* tab.
3. Select the *By Simulator* radio button and select your simulator in the drop-down list.
4. In the following fields, you can specify how you want the software to netlist your component, what parameters to pass to the underlying subcircuit or model definition, and the order of the input terminals (for netlisting):

Field	Value
netlistProcedure	Type the name of a netlist procedure or leave this field blank to use the default netlist procedure.
otherParameters	Type `model` to specify a parameter for model or subcircuit name.
instParameters	Type the names of one or more component parameters (space-separated) you want to pass into the subcircuit or model.
termOrder	Type the names of the symbol's terminals in the order you want them netlisted (the order must match the node order on the `subckt` line or that of the model referenced).

Note: For more information, see "Modifying Simulation Information" in the *Component Description Format User Guide*.

Creating a Stopping Cellview

Important

You must have write privileges on the library to perform this task.

To create a stopping cellview for your simulator, do the following:

1. Edit the symbol cellview.

2. Choose *Design – Save As*.

   The Save As form appears.

3. In the *View Name* field, type a view name to represent your simulator.

   For example, for the Spectre circuit simulator, type `spectre`.

4. Click *OK*.

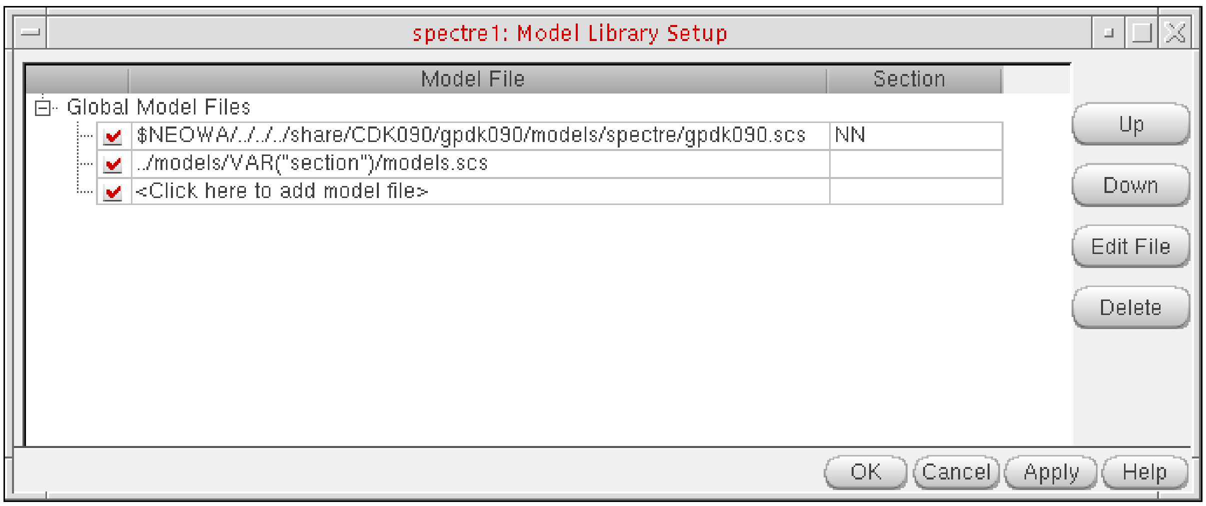
   The environment creates a new cellview (as a directory of files) in the specified library.

Varying the Model File and Section during Simulation

To make the model file or section something you can vary during simulation, do the following:

1. On the Data View assistant pane, right-click the test or analysis name and choose *Model Libraries*.

   The Model Library Setup form appears.



2. In the *Model File* column, double-click and type a variable name for the model file you want to vary (or as a part of the model file path) using the following format:

   `VAR("`*`variableModelFileName`*`")`

   For example:

   `VAR("myModelFile")`

   or

   `../models/VAR("section")/models.scs`

3. In the *Section* column, double-click and type a variable name for the section you want to vary. For example:

   `VAR("myModelCorner")`

   Note: For more information about sections in model files, see "Corners Modeling" in the Direct Simulation Modeling User Guide.

4. Click *Apply*.

   Variables you specify using `VAR` (such as *myModelFile* or *section*, and *myModelCorner*) appear on the Variables tab of the Variables and Parameters assistant pane.

5. Double-click in the *Value* column for each global variable and type a value. For example:

Name	*Value*
Global Variables	
myModelCorner	`"NN"`
myModelFile	`"$PROJECT1/path/to/myModels.scs"`

You can vary these global variables in parametric sweeps and corners analysis.

Here is an example application using `VAR` in the specification of a model file to sweep through different model types.

If you have model files stored in a directory hierarchy such as the following:

`../models/`*`sectionIdentifier`*`/models.scs`

where *`sectionIdentifier`* is the model type (such as `typ`, `ff`, `ss`), you can create a sweep of model types as follows:

1. On the Model Library Setup form, double-click in the *Model File* column and type the following:

   ```
   ../models/VAR("section")/models.scs
   ```

2. Click *OK*.

   The *section* variable appears in the *Global Variables* tree on the Variables tab of the Variables and Parameters assistant pane.

3. Double-click in the *Value* column for *section* and click the ellipsis that appears at the right end of the field.

   The Parameterize form appears.

4. Click Add Specification and select *Inclusion List* from the drop-down list:

5. In the *Values* field, type the set of model types (valid section names from the model file) through which you want the program to sweep.

   For example:

   ```
   "typ", "ff", "ss"
   ```

   The program sweeps through all values of `section`.

Editing a Model File

To edit a model file, do the following:

1. On the Data View assistant pane, right-click the test or analysis name and choose *Model Libraries*.

   The Model Library Setup form appears.

2. Select the model library you want to edit and click *Edit File*.

   The model file appears in an editing window.

3. Make the edits you want to make.
4. Save and exit the file.
5. On the Model Library Setup form, click *OK*.

Disabling a Model File

To disable a model file for a test, do the following:

1. On the Data View assistant pane, right-click the test or analysis name and choose *Model Libraries*.

   The Model Library Setup form appears.

2. Deselect the check box for the model library you want to disable.
3. Click *OK*.

10

Netlisting

The environment generates the following files for simulation:

File	Description
Netlist	Contains component and design connectivity information only
Simulator input file	Contains the netlist and simulator control information

Note: The environment passes the simulator input file to your configured simulator.

ADE XL creates or updates the simulator input file automatically when you give the command to run a simulation. The environment creates hierarchical netlists incrementally: incremental netlisting is faster than full hierarchical netlisting because the environment updates the netlist only for those schematics that have changed since it created the previous netlist.

Alternatively, you can generate a netlist and simulator input file when

- You want to run the simulator in stand-alone mode
- You want to modify the netlist to take advantage of features that the design environment interface to your simulator does not support
- You want to read the netlist before running the simulation

See the following topics for more information:

- Creating a Netlist on page 364
- Displaying a Netlist on page 364
- Expanding Hierarchy to Netlist a Design on page 365

Creating a Netlist

To create and display the simulation input file, which contains the netlist, do the following:

➤ On the Data View assistant pane, right-click the test or analysis name and choose *Netlist – Create*.

To recreate the netlist, choose *Netlist – Recreate.*

Displaying a Netlist

To display an existing simulator input file, do one of the following:

➤ On the Data View assistant pane, right-click the test and choose *Netlist – Display*.

➤ On the Results tab of the Outputs pane, right-click a test and choose *Netlist – Display*.

Expanding Hierarchy to Netlist a Design

While netlisting a hierarchical design, the environment expands every cell (instance) into lower level cells until it reaches one designated as a primitive. The environment adds each primitive to the netlist.

At each level in your design hierarchy, you can have one or more views for each cell. You use a view list to specify which view the design environment selects for expansion during netlisting. View lists can be global to the entire design or specific to an instance as specified by its property values.

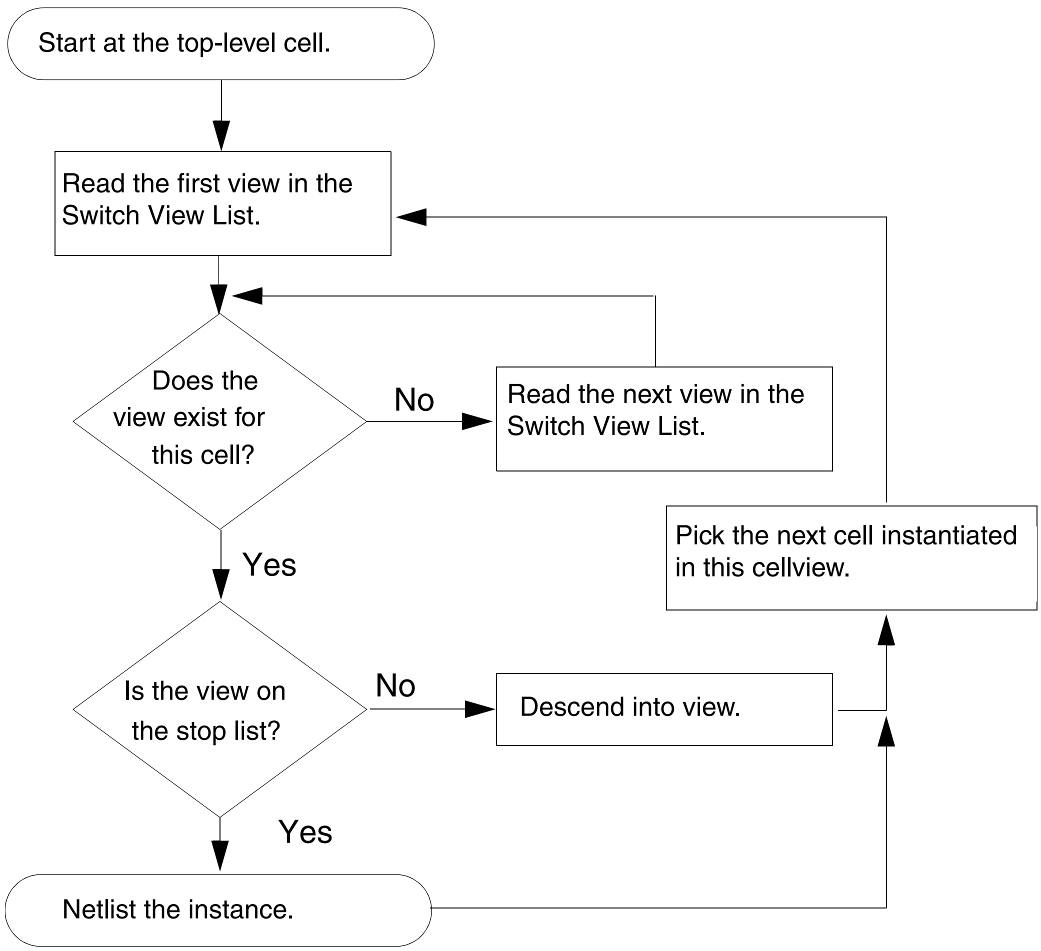
For analog simulation, you specify the global or default view list in the *Switch View List* field of the Environment Options form when the selected cellview is not a configuration (config) view. If an instance does not have any of the views listed in the switch view list, the netlister reports an error.

The netlister uses the stop list, which you specify in the *Stop View List* field on the Environment Options form, to identify primitives. When the netlister reaches a view that is specified in both the switch and the stop list, it netlists the instance and does not expand beyond this level.

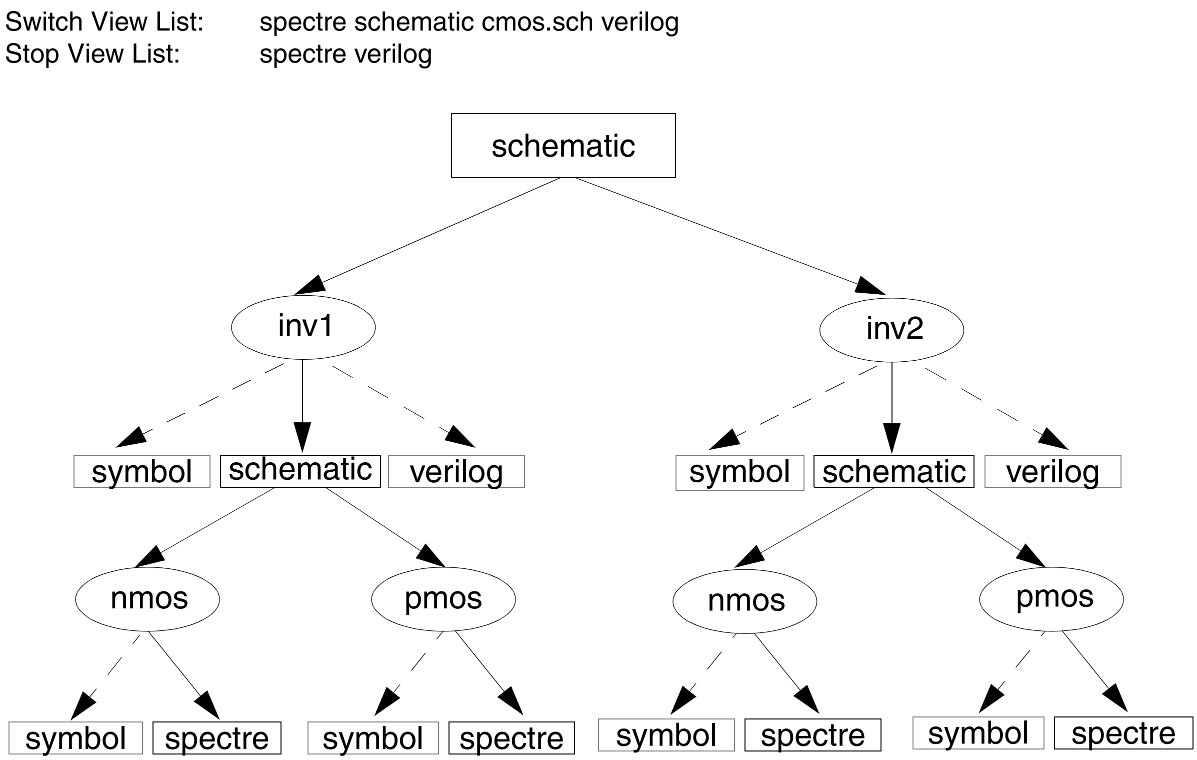
For more information, see the *Virtuoso AMS Hierarchy Editor User Guide*.

Note: Parasitic simulation and mixed-signal simulation use different processes for creating switch and stop view lists. See the *Virtuoso Parasitic Aware Design User Guide* and the Virtuoso Mixed-Signal Circuit Design Environment User Guide for more information.

The following flowchart shows how a typical OSS-based netlister netlists a design.



The following figure illustrates how hierarchy expansion is performed on a simple design. The solid lines show the view selection and design expansion based on the switch view list and stop view list shown.



11

Selecting Data to Save and Plot

From the Outputs Setup tab of the Outputs pane, you can specify nets, terminals, and measurements you want to save and plot. You can save all node voltages and terminal currents or specify a set of voltages and currents you want to save. You can select output nodes and terminals on your schematic and build expressions using the Calculator to analyze or measure particular results.

Each item that appears on the Outputs Setup tab has a *Plot* check box and a *Save* check box. You can select the check box for those outputs you want to plot or save, or deselect the check boxes of those items you do not want to plot or save.

See the following topics for more information:

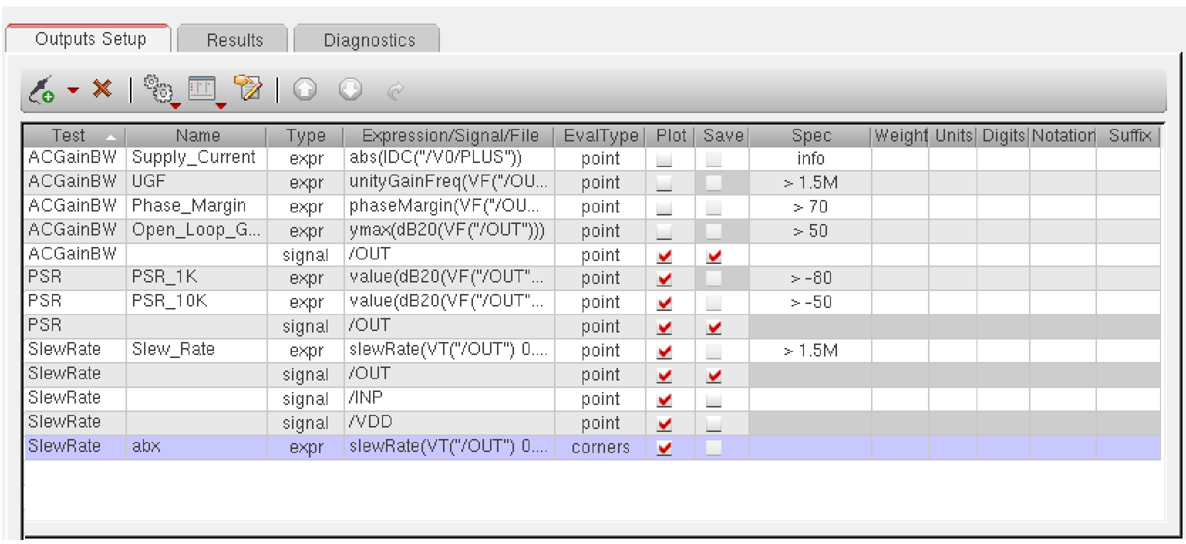
Opening the Outputs Setup Tab

You can use the Outputs Setup tab of the Outputs pane to add a new output signal/ expression/OCEAN or MATLAB script or to change an existing output.

To open the *Outputs Setup* tab, do the following:

➤ On the Welcome to ADE XL pane, click the hypertext link below *Outputs*.

The Outputs pane appears displaying the *Outputs Setup* tab.



See the following topics for more information:

- Selecting Outputs on the Schematic on page 372
- Adding an Output Expression on page 376
- Modifying an Output Expression on page 391
- Specifying Whether a Result Will Be Saved or Plotted on page 375
- Copying Outputs on page 396
- Adding User-Defined Columns in the Outputs Setup Tab on page 398
- Configuring How Outputs Appear on the Outputs Setup Tab on page 407
- Selecting Outputs to Save or Plot on page 410
- Removing Outputs on page 410
- Loading an OCEAN or a MATLAB Measurement on page 392

Selecting Outputs on the Schematic

You can select outputs on the schematic and specify whether you want to save or plot them.

To select an output on the schematic, do the following:

Note: This procedure allows you to select only one output on the schematic. If you want to select multiple outputs on the schematic at a time, see Selecting Outputs to Save on page 373 and Selecting Outputs to Plot on page 373.

1. On the Outputs Setup tab, click the  button.
2. In the drop-down list, select a test and choose *Signal*.

   A new row is added for the test with the output type *signal*.

Tip

You can double-click on the test name or output type to select a different test or output type from the drop-down list that appears in the *Test* or *Type* column.

3. (Optional) In the *Name* field, type the name for the output.

   This name appears in the Waveform window.
4. Double-click on the *Expression/Signal/File* field and click the ellipsis  button.

   The schematic window appears in a tab.
5. In the schematic, select a signal.

   The signal name appears in the *Expression/Signal/File* field on the Outputs Setup tab. By default, the *Plot* and *Save* check boxes are selected.
6. (Optional) You can select or deselect either or both of the check boxes:
   - ❑ *Plot* — The program plots the specified signal output after simulation.
   - ❑ *Save* — The program saves the signal output to the simulation results file.

See also the following topics for more information:

- Selecting Outputs to Save on page 373
- Selecting Outputs to Plot on page 373
- Selecting Nodes, Nets, and Terminals on page 374
- Setting Plotting Options for Specific Tests on page 620

Selecting Outputs to Save

To select outputs to be saved, do the following:

1. On the Outputs Setup tab of the Outputs pane, right-click a test and choose *To be saved*.

   The schematic window appears in a tab.

2. On the schematic, select one or more nodes, nets, or terminals.
3. Press the *Esc* key when you are done.

   The name of each selected item appears (such as */out* or */I2/PLUS*) in different rows in the Outputs Setup tab. The *Save* check box is selected.

Selecting Outputs to Plot

To select outputs to be plotted, do the following:

1. On the Outputs Setup tab of the Outputs pane, right-click a test and choose *To be plotted*.

   The schematic window appears in a tab.

2. On the schematic, select one or more nodes, nets, or terminals.
3. Press the *Esc* key when you are done.

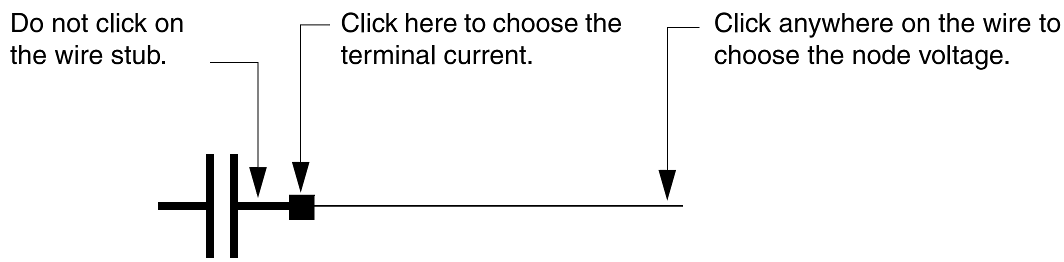
   The name of each selected item appears (such as */out* or */I2/PLUS*) in different rows in the Outputs Setup tab. The *Plot* check box is selected.

   Note: For nets, both the *Plot* and *Save* check boxes are selected.

   The program plots the selected items in a waveform window at the end of the simulation.

See also Setting Plotting Options for Specific Tests on page 620.

Selecting Nodes, Nets, and Terminals



A circle appears around each pin when you choose a terminal and wires appear highlighted when you choose a net. You can alternate between selecting a terminal and selecting the wire to which it is connected.

You can

- Click on an instance to choose all instance terminals.
- Click on the square pin symbols to choose currents.
- Click on wires to choose voltages.
- Click and drag to choose voltages by area.
- Select nodes and terminals on lower-level schematics by doing the following:
  - **a.** In the schematic window, choose *Design – Hierarchy – Descend Edit*.
  - **b.** Select a hierarchical instance.

    The Descend form appears.
  - **c.** Click *OK*.

    The lower-level schematic appears.
  - **d.** In the schematic window, select one or more nodes, nets, or terminals.

Specifying Whether a Result Will Be Saved or Plotted

You can specify whether an item will be saved to the results database or plotted in a waveform window on the Outputs Setup tab of the Outputs pane by selecting or deselecting the *Plot* or *Save* check box for the item. Selecting the *Plot* check box causes the selected item or items to appear in a waveform window at the end of the simulation.

Note: If none of the outputs specified for a test are selected for plotting, the following warning message is displayed in the CIW:

```
(ADEXL-1617): Following tests do not have any outputs selected for plotting:
<testname1>
<testname2>
...
```

Adding an Output Expression

You can use the Outputs Setup tab of the Outputs pane to add output expressions. You can type an output expression directly in the *Expression/Signal/File* field or use the Calculator to build an output expression.

To add an output expression, do the following:

1. On the Outputs Setup tab, click .
2. In the drop-down list, select a test and choose *Expression* from the sub menu.

   A new row is added for the test with the output type *expr*.

Tip

You can double-click on the test name or output type to select a different test or output type from the drop-down list that appears in the *Test* or *Type* column.

3. (Optional) In the *Name* field, type a name for the expression.

   This expression name appears in the Virtuoso Visualization and Analysis XL window.

4. (Optional) The default value in the *EvalType* column is `point`. This signifies that by default the expression will run for the given design point. If you want to run the expression for all corners across the design point, double-click in the *EvalType* cell and select `corners` from the pull-down list.

   When you set the *EvalType* as `corners`, the color of the row is changed to blue. For more details, refer to Creating Expressions to be Measured Across Corners on page 382.

5. Double-click on the *Expression/Signal/File* field.
6. Type an output expression directly in the *Expression/Signal/File* field or use the Calculator to build an output expression. To add the output expression using the calculator, do the following:

   a. Click the ellipsis  button.

      The Calculator window appears.

   b. In the Calculator window, build an output expression.

      For example, you can build an expression for the 3 dB point of an output signal as follows:

```
bandwidth(VF("/OUT), 3, "low")
```

c. Do one of the following:

❍ Choose *Tools – Send to ADEXL Test* in the Calculator window.

❍ On the Outputs Setup tab, click the button in the *Expression/Signal/File* field.

The expression appears in the *Expression/Signal/File* field.

Note: While performing the division of two integer or floating point values, if denominator is greater than numerator, the division results are truncated. For example, if you perform the division of 1/2, the int function displays the result as 0 instead of 0.5.

For more information about how the integer and floating point division is performed using SKILL, see Integer vs. Floating-Point Division in the Arithmetic and Logical Expressions chapter of the *Cadence SKILL Language User Guide*.

See also:

❑ Creating Dependent Expressions on page 379

❑ Creating Expressions to be Measured Across Corners on page 382

7. (Optional) Change the *Plot* and *Save* options for the expression, if required. See also Specifying Whether a Result Will Be Saved or Plotted on page 375.

8. (Optional) In the *Spec* and *Weight* columns next to the expression, define the specification and the weighting factor for the specification for the output expression. For more information, see Defining a Specification on page 703.

9. (Optional) In the *Units* column next to the expression, specify the unit value to be used for displaying the measured results for the expression in the Results tab of the Outputs pane.

For example, if you specify the unit value as `dB` for an expression named `Gain`, the measured results for the expression is displayed with the unit value `dB` in the Results tab of the Outputs pane.

10. (Optional) In the *Digits* column next to the expression, specify the number of significant digits in which the measured results for the expression needs to be displayed in the Results tab of the Outputs pane. Valid values are `2` to `15`.

Note: If you do not specify the number of significant digits, the default number of significant digits specified in the Default Formatting Options form will be used to display the measured results for the expression in the Results tab. For more information, see Specifying Default Formatting Options on page 563.

11. (Optional) Double-click in the *Notation* column next to the expression. From the drop-down list that appears, choose the notation style in which the measured results for the expression needs to be displayed in the Results tab of the Outputs pane.

Choose	To
default	Display results using the default notation style specified in the Default Formatting Options form. For more information, see Specifying Default Formatting Options on page 563.
eng	Display results in the engineering notation.
sci	Display results in the scientific notation.
suffix	Display results in the suffix notation. If you choose suffix, double-click in the *Suffix* column next to the expression and choose the suffix from the drop-down list. If you specify the suffix as *auto* or do not specify the suffix in the Suffix column, the suffix is automatically assigned.

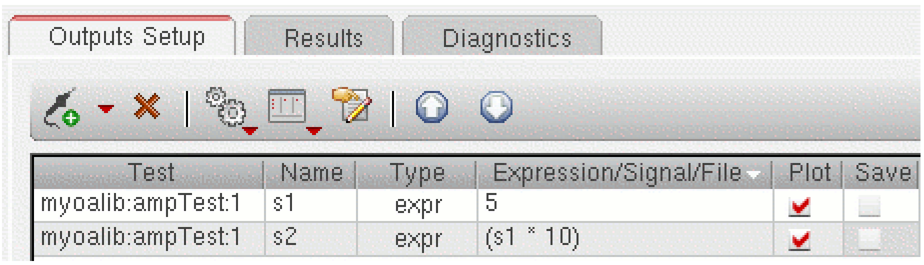
Note: If you do not specify the notation style, the default notation style specified in the Default Formatting Options form will be used to display the measured results for the expression in the Results tab. For more information, see Specifying Default Formatting Options on page 563.

See also:

■ Creating Dependent Expressions on page 379
■ Creating a Combinatorial Expression on page 381
■ Creating Expressions to be Measured Across Corners on page 382
■ Modifying an Output Expression on page 391
■ Loading an OCEAN or a MATLAB Measurement on page 392
■ Setting Plotting Options for Specific Tests on page 620
■ Working with Specifications on page 701
■ Specifying Default Formatting Options on page 563

Creating Dependent Expressions

You can create expressions based on other expressions. For example, assume that you have an expression named `s1` with the expression `5`. If you want another expression, say, `s2` to be ten times the value of `s1`, type `s1*10` in the *Expression/Signal/File* field as shown in the figure below.



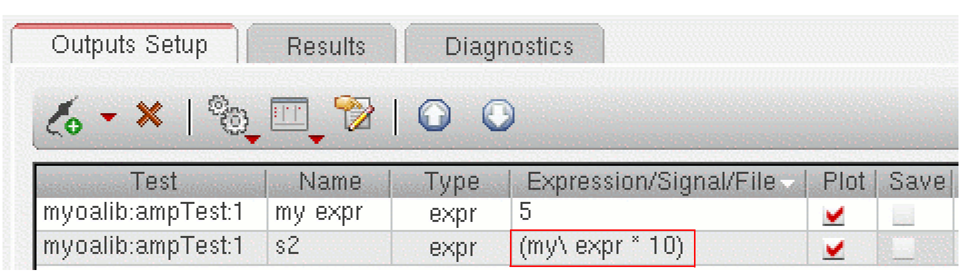
Note the following when you create dependent expressions:

- Expressions can be added in any order, irrespective of their dependencies. For example, if the expression `s2` is based on expression `s1`, it is not necessary to add the expression `s1` before adding the expression `s2`.
- An expression can be based on any number of other expressions.
- Ensure that there is no cyclic dependency between the dependent expressions.

  In the following example, a cyclic dependency exists because expression `myExpr` depends on expression `myExpr1`, and expression `myExpr1` also depends on expression `myExpr`.

  ```
  myExpr=myExpr1*5
  myExpr1=myExpr+10
  ```

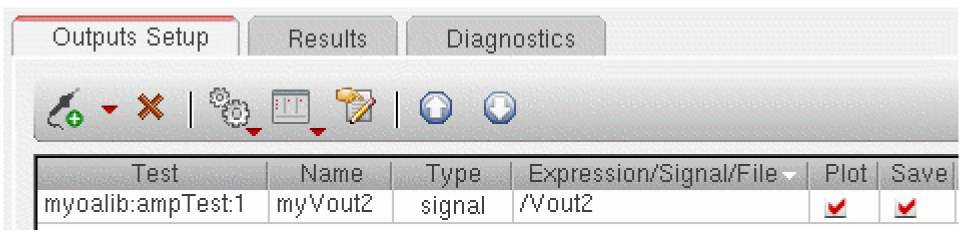
- If the expression name that is being used in a dependent expression has special characters such as spaces or dot (.), use the \ character to escape these special character. For example, if you are using an expression named `my expr` in another expression, escape the space character in `my expr` as shown in the figure below:



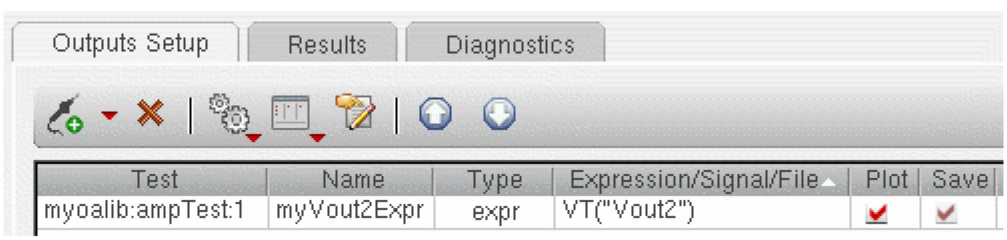
Note: Generally, expression names can have special characters as they are valid string values. However, when these expression names are used in dependent expressions, they are treated as SKILL symbols and should be in a valid SKILL format. Therefore, special characters in such expression names should be escaped using the \ character.

- It is not necessary to plot or save the expressions on which other expressions are based. For example, if the expression `s2` is based on expression `s1`, it is not necessary to select the *Plot* or *Save* check box for the expression `s1` in the Outputs Setup tab.

- You cannot assign a name to an output of type signal and then use that name in a dependent expression. For example, you cannot assign the name `myVout2` for the `Vout2` signal as shown below, and then use `myVout2` in a dependent expression.



If you want a dependent expression to be based on a signal, create an expression based on that signal and use the name of that expression in a dependent expression. For example, create an expression named `myVout2Expr` based on the signal `Vout2` as shown below, and then use `myVout2Expr` in a dependent expression.



- Error messages, if any, regarding dependent expressions are displayed in the Command Interpreter Window (CIW).

Creating a Combinatorial Expression

A combinatorial expression is one created using more than one output from one or more tests. For example, you might want to create an expression such as

`Test1_Output1/Test2_Output2`

To create a combinatorial expression such as the one above, do the following:

1. On the Outputs Setup tab, click the button.
2. In the drop-down list, select a test and choose *Expression*.

   A new row is added for the test with the output type *expr*.

Tip

You can double-click on the test name or output type to select a different test or output type from the drop-down list that appears in the *Test* or *Type* column.

3. (Optional) In the *Name* field, type the name for the expression (such as `Out1DivOut2`).

   This name appears in the Waveform window.
4. Double-click on the *Expression/Signal/File* field.
5. Click the ellipsis button.

   The Calculator window appears.
6. On the Outputs Setup tab, drag an output from any test and drop it in the buffer on the Calculator window.
7. Type `/` at the end of the expression.
8. On the Outputs Setup tab, drag another output from another test and drop it at the end of the expression in the buffer on the Calculator window.
9. Do one of the following:
   - ❑ Choose *Tools – Send to ADEXL Test* in the Calculator window.
   - ❑ On the Outputs Setup tab, click the button in the *Expression/Signal/File* field.

   The expression appears in the *Expression/Signal/File* field.

Note: The expression uses the `calcVal` function.

Creating Expressions to be Measured Across Corners

You can create expressions that measure results across all corners for a design point.

Important

Currently, this feature is supported in the following run modes:
`- Single Run, Sweeps, and Corners`
`- Monte Carlo`
`- Local Optimization`
`- Global Optimization`

To evaluate expressions across corners, perform the following steps:

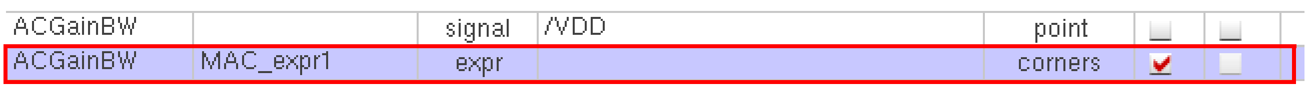
1. Right-click in the *Outputs Setup* tab and choose *Add Expression*.

   A new blank output row with output type `expr` is added to the outputs table.

2. In the new row, double-click in the *Test* column and select a test name.

3. Double-click in the *EvalType* column and specify the evaluation type as `corners`.

   The color of the row changes to blue.



   Note: Outputs of type Signal, Matlab script, Area Spec, Device Check, and Op Region Spec are always evaluated for design points, that is, when evalType is equal to `point`.

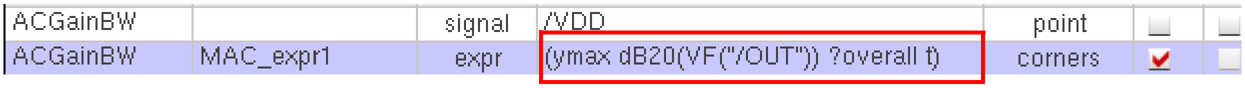
4. (Optional) Add a name for the expression in the *Name* column.

5. Enter an expression in the *Expression/Signal/File* column.

   When the evaluation type os set as `corners`, if you use any of the following functions in the expression, the tool automatically adds an additional argument, `overall`, to the function call and sets it to `t`:

   - average
   - ymax
   - ymin
   - peakToPeak
   - stddev

An example is shown in the following figure.



When the argument `overall` is set to `t`, it performs the calculation on the results of corner simulations for each design point that are treated as discrete values for evaluation and not waveforms.

For more details on how the measures are calculated across corners, refer to Calculations of Measurements Across Corners.

Caution

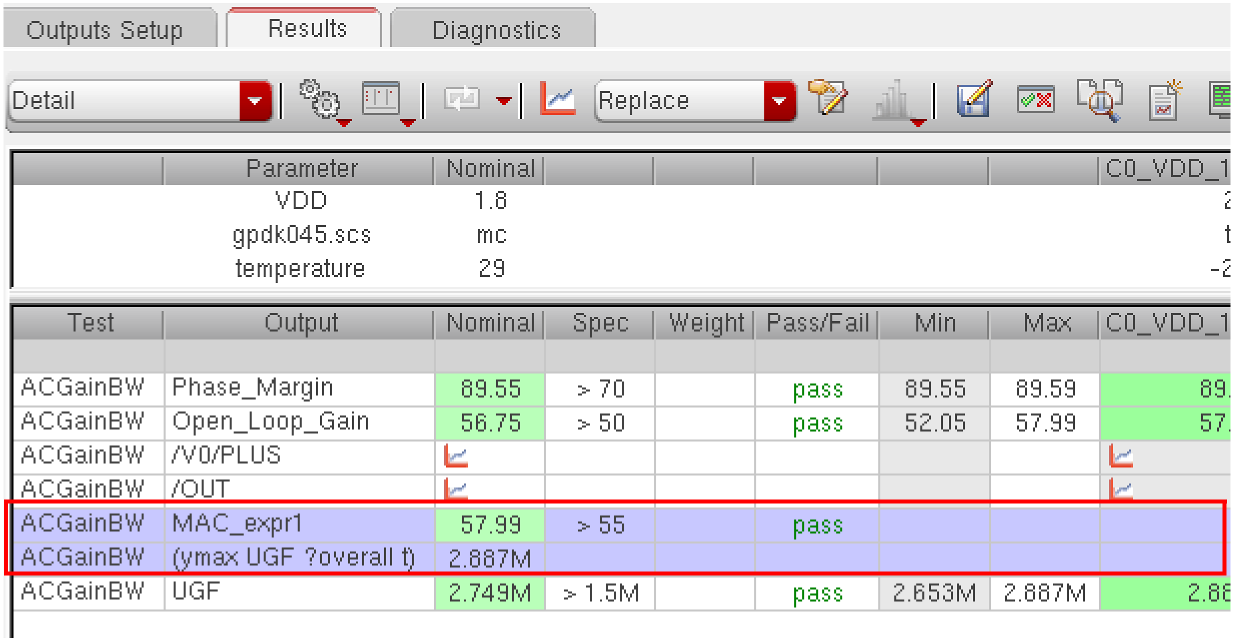
The new argument, overall, has been added to only five most-frequently used Calculator functions listed above. Usage of this argument in any other function might generate erroneous results.

There are some more ways in which you can create expressions to be measured across corners. For more details, refer to Alternate Ways to Create Measurements Across Corners on page 385.

6. (Optional) Add specification details in the *Spec* column.

   Note: The *Override Specification* command in the popup menu on the Output Setup tab is disabled for measurements across corners.

When you run simulations, results of the expressions measured across corners are displayed on the *Results* tab, as shown in the following figure.



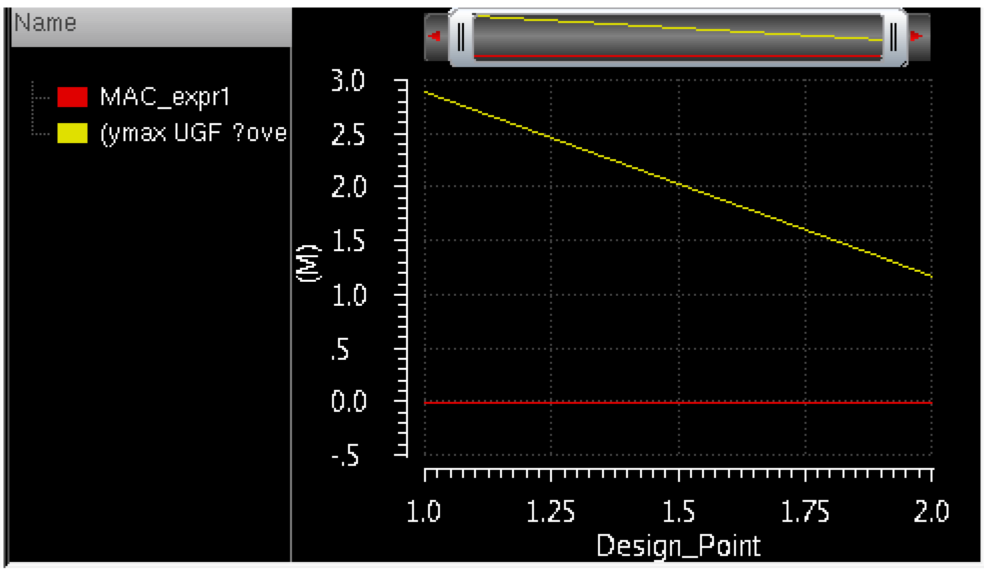
Note the following on the *Results* tab:

- Rows that display results of the expressions measured across corners appear in blue.
- Results of the expressions measured across corners are displayed in the *Nominal* column and the columns for other corners are blank. No new column is created to show this result.
- If you disable the nominal corner before running simulation, the results are still displayed in the *Nominal* column. Only the cells that display results for expressions measured across corners show results. Other cells in the column do not have any results.
- If you specify performance specifications for the expressions measured across corners, depending on the measured value, the tool displays the pass/fail status.
- If you troubleshoot a design point, ADE XL ignores the outputs that are measured across corners.
- If you open the debug environment, the outputs that are measured across corners are not shown in the *Outputs* section of the debug environment.
- Re-evaluation is not supported for the outputs that are measured across corners.
- For Monte Carlo simulations, if you clear the *Save Data to Allow Family Plots* check box and if you have outputs to be measured across corners as independent expression then it will not be evaluated and show eval error. This is because the

evaluation is based on simulation results that are not saved. However, note that if an expression to be measured across corners is fully dependent on other expressions for which results are saved in the results database, it is evaluated and results are shown.

Also note that:

- ❑ When you plot the results using Auto Plot, the results of measurements across corners are plotted across all design points, as shown in the following figure.



- ❑ The results of measurements across corners are displayed in all views on the *Results* tab.

- ❑ For Monte Carlo results, in the yield view, only a single row is displayed for the expressions measured across corners. However, for other expressions, result for each corner is displayed in a separate row.

Alternate Ways to Create Measurements Across Corners

Alternate ways in which you can create expressions to be evaluated across corners are:

- By creating OCEAN measures in a .ocn file and loading that file. For example:

```
MAC_ocn1= ymax(bandwidth(dB20(mag(VF("/OUT"))) 10 "low") ?overall t)
axlOutputResult(MAC_ocn1 "MAC_ocn1")
```

While creating an OCEAN measure in one file, you can also use an expression defined in another OCEAN file. For example, ocean1.ocn is defined as given below.

```
BW=bandwidth(VF("/net1") 10 "low")
axlOutputResult(BW "BW")
```

In ocean2.ocn, you can create another measure that uses BW that was defined in ocean1.ocn. This is shown in the example given below.

```
MAC_ocn1=ymax(calcVal("BW" "testLib:top:1") ?overall t)
axlOutputResult(MAC_ocn1 "MAC_ocn1")
```

Important

If you want to create an OCEAN measure of both the types, point and corners, you need to create separate OCEAN script files where one file contains functions of a single type only. Use a separate *`axlOutputResult`* statement for each output.

- By using the `ocnxlOutputExpr` OCEAN function in an OCEAN script to create an output. This function supports the `evalType` argument that specifies how to evaluate the expression.

Calculations of Measurements Across Corners

When the *evalType* column is set to `corners`, the measurements are calculated across corners. This section describes how different scalar and waveform expressions are calculated to measure the outputs across corners.

Measurements of Scalar Expressions

When the *evalType* column is set to `corners`, by default the *`overall`* argument is set to `t`. With this default value of *`overall`*, the measurement is calculated by using the result values obtained for each corner, irrespective of the number of sweep variables. This gives a scalar value as result.

In the example shown below, **average** of risetime is calculated by adding the risetime calculated for all the sweep variables and then dividing the sum by the count of values.

rt_avg = average of risetime scalar values calculated for each corner

See the results of rt_avg in the figure given below.

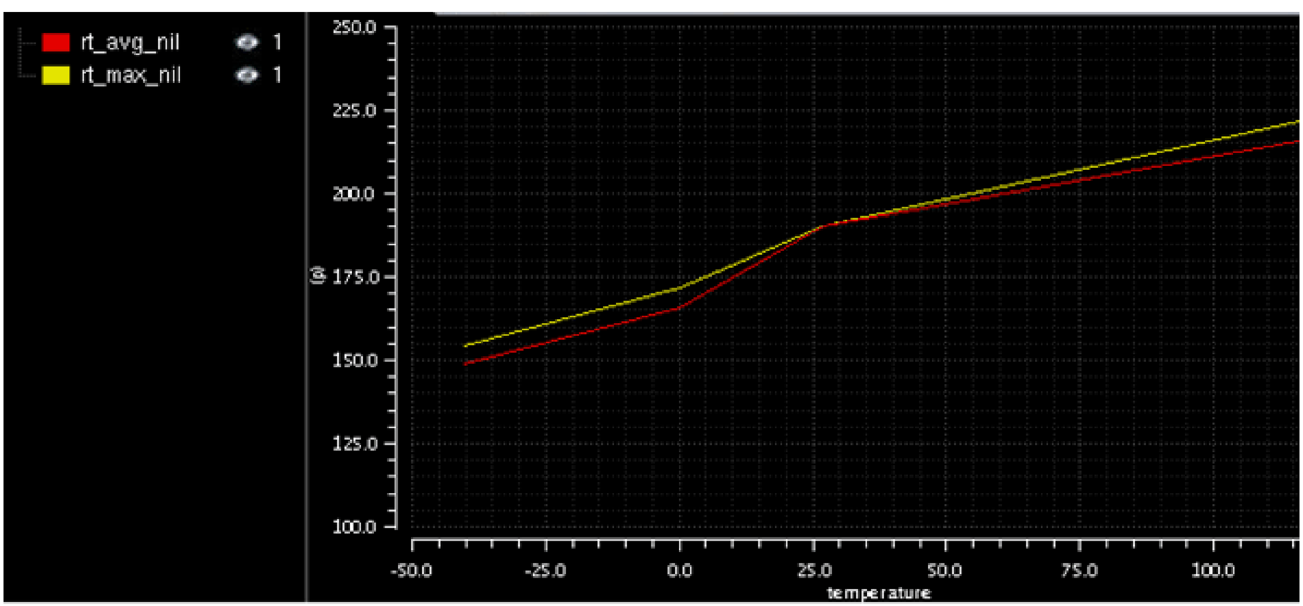
TRAN	risetime	expr	riseTime(VT("/OUT") 0 t 1e-08 t 10 90 nil "time")	point
TRAN	rt_avg_t	expr	(average risetime ?overall t)	corners
TRAN	rt_avg_nil	expr	(average risetime ?overall nil)	corners
TRAN	rt_max_t	expr	(ymax risetime ?overall t)	corners
TRAN	rt_max_nil	expr	(ymax risetime ?overall nil)	corners

Parameter	Nominal	C0_0	C0_1	C0_2	C0_3	C0_4	C0_5	C0_6	C0_7
temperat...	27	-40	0	125	-40	0	125	-40	0
vdd	1.8	1.9	1.9	1.9	2	2	2	2.1	2.1

Test	Output	Nominal	C0_0	C0_1	C0_2	C0_3	C0_4	C0_5	C0_6	C0_7
TRAN	risetime	190.2p	154.7p	171.7p	225.1p	148.7p	165.6p	218p	143.6p	160.3p
TRAN	rt_avg_t	178.9p								
TRAN	rt_avg_nil									
TRAN	rt_max_t	225.1p								
TRAN	rt_max_nil									

If you change the default value of the `overall` argument and set it to `nil`, this returns a scalar value in case of a single sweep variable, but a waveform output in case of multiple sweep variables.

In the example given above, for rt_max_nil, ymax is calculated across vdd at each temperature value and a waveform is plotted for each temperature. The result is shown in the figure given below.



Note: Getting a waveform in case of multiple sweep variables is not the desired output for measurements across corners.

Measurements of Waveform Expressions

For the default value of the *overall* argument, a measurement is first calculated on the waveform for each corner thereby resulting into a set of values. The measurement is further applied on this resulting set, thereby giving a scalar value as a result.

Consider the example shown below.

TRAN	avg	expr	average(VT("/OUT"))	point
TRAN	wave_max	expr	ymax(VT("/OUT"))	point
TRAN	wave_min	expr	ymin(VT("/OUT"))	point
TRAN	wave_avg_t	expr	(average VT("/OUT") ?overall t)	corners
TRAN	wave_avg...	expr	(average VT("/OUT") ?overall nil)	corners
TRAN	wave_pp_t	expr	(peakToPeak VT("/OUT") ?overall t)	corners

	Parameter	Nominal			C0_0	C0_1	C0_2	C0_3	C0_4	C0_
	temperature	27			-40	0	125	-40	0	12
	vdd	1.8			1.9	1.9	1.9	2	2	2

Test	Output	Nominal	Min	Max	C0_0	C0_1	C0_2	C0_3	C0_4	C0_
TRAN	avg	717.2m	717.2m	843.8m	762.4m	760.5m	752.5m	803.2m	801.8m	794.
TRAN	wave_max	902.5m	902.5m	1.055	952.4m	952.7m	953.5m	1.003	1.003	1.00
TRAN	wave_min	-623.6m	-657.5m	-614.4m	-614.4m	-628m	-637.6m	-624.4m	-638m	-647.
TRAN	wave_avg_t	791.4m								
TRAN	wave_avg_t1									
TRAN	wave_pp_t	1.712								

In this example, for **wave_avg_t**, first average of `VT("/OUT")` is calculated for each corner individually and then discrete average of the resulting values is taken, as shown below.

```
average = average values obtained for corners
```

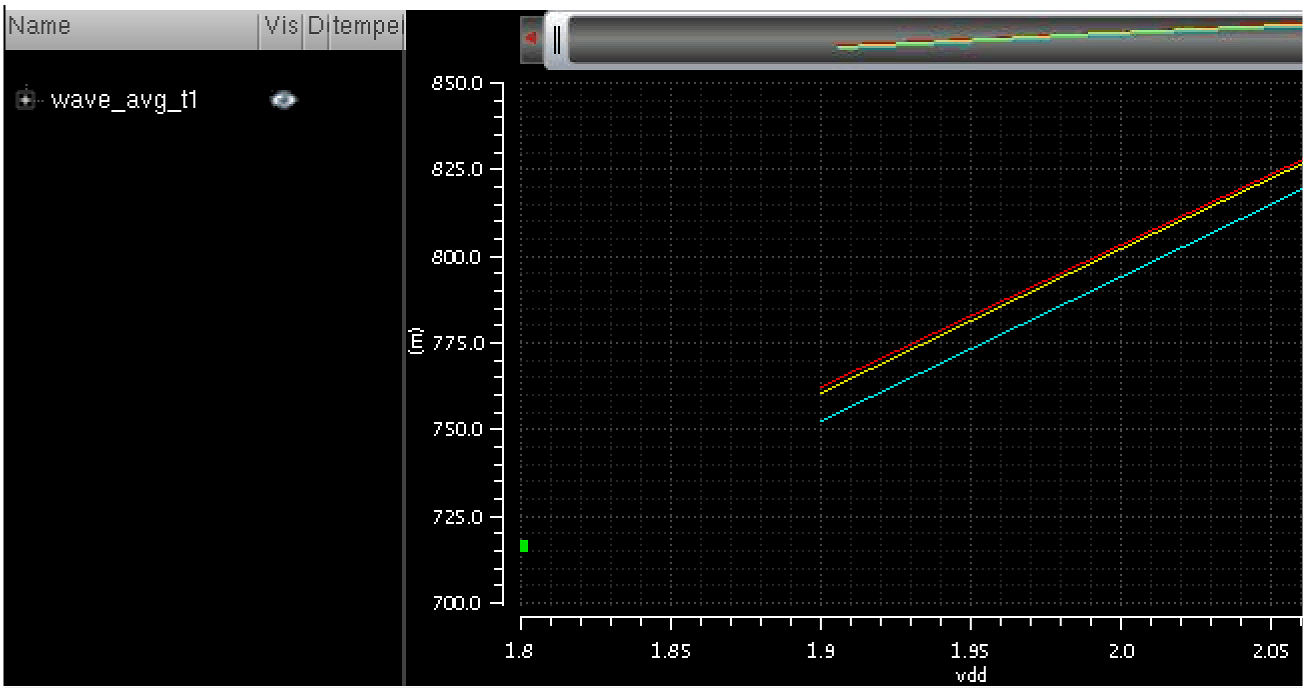
Similarly, for **wave_pp_t**, first, ymax and ymin are calculated for `VT("/OUT")` for each corner. Next, ymax and ymin of the resulting values is calculated.

```
ymax_all = ymax of the ymax values of individual waveforms
ymin_all = ymin of the ymin values of individual waveforms
```

Next, peakToPeak is calculated using the resulting ymax_all and ymin_all values, thereby giving a scalar result.

```
peakToPeak = ymax_all - ymin_all
```

However, if you set the *`overall`* argument to `nil`, a measurement is calculated on the waveform for each corner and the resulting values are plotted. This is not the expected result for measurements across corners. In the example shown above, for wave_avg_t1, average is calculated for each combination of VDD and temperature and the resulting values are plotted, as shown in the figure shown below.

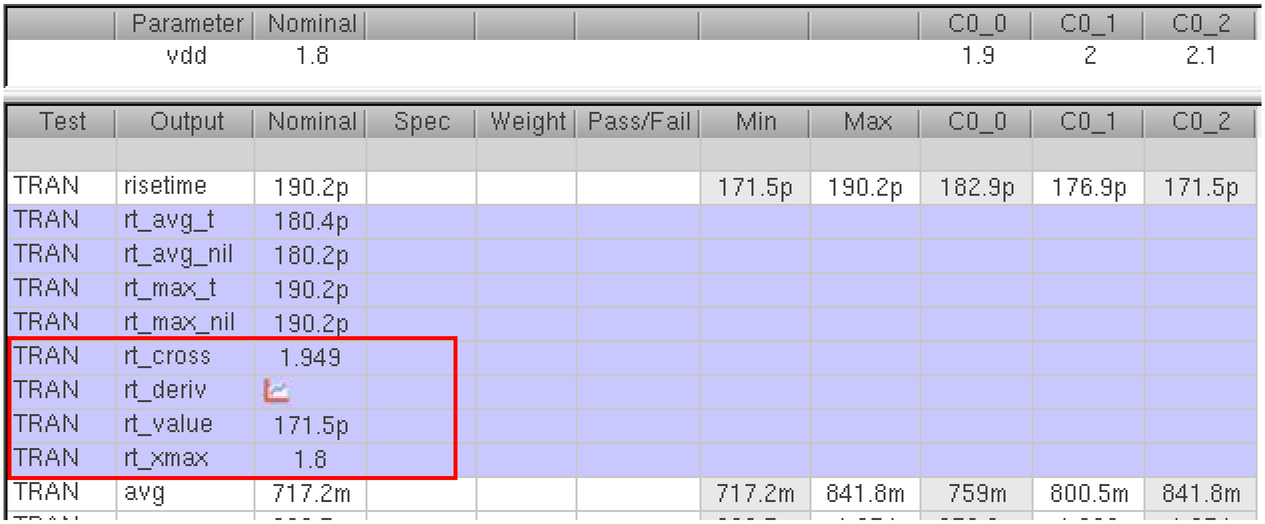


Measurements Across Corners for Functions that Run on a Single Corner Variable

Some of the calculator functions that do not support the new argument, `overall`, give correct results when calculated across corners. For example, functions such as `xmin`, `xmax`, `deriv`, `cross`, and `value`, that work on a single dimension, give correct results when calculated across corners, as shown below.

TRAN	rt_cross	expr	cross(risetime 1.8e-10 1 "falling")	corners
TRAN	rt_deriv	expr	deriv(risetime)	corners
TRAN	rt_value	expr	value(risetime "vdd" 2.1)	corners
TRAN	rt_xmax	expr	xmax(risetime)	corners

The results for these outputs are shown below.



	Parameter	Nominal						C0_0	C0_1	C0_2
	vdd	1.8						1.9	2	2.1

Test	Output	Nominal	Spec	Weight	Pass/Fail	Min	Max	C0_0	C0_1	C0_2
TRAN	risetime	190.2p				171.5p	190.2p	182.9p	176.9p	171.5p
TRAN	rt_avg_t	180.4p								
TRAN	rt_avg_nil	180.2p								
TRAN	rt_max_t	190.2p								
TRAN	rt_max_nil	190.2p								
TRAN	rt_cross	1.949								
TRAN	rt_deriv									
TRAN	rt_value	171.5p								
TRAN	rt_xmax	1.8								
TRAN	avg	717.2m				717.2m	841.8m	759m	800.5m	841.8m

However, it is not recommended to measure outputs across corners for other functions that work on multiple dimensions and do not support the `overall` argument.

Modifying an Output Expression

You can use the Outputs Setup tab of the Outputs pane to modify output expressions. You can modify an output expression directly in the *Expression/Signal/File* field or use the Calculator to modify an output expression.

To modify an output expression, do the following:

1. On the Outputs Setup tab, double-click on the expression you want to change in the *Expression/Signal/File* field.

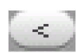
2. Make your changes by modifying what appears in the *Expression/Signal/File* field.

   You can make your changes by typing them directly in the *Expression* field or by opening a Calculator window. To modify the output expression using the calculator, do the following:

   a. Click the ellipsis  button.

      The Calculator window appears.

   b. In the Calculator window, change the output expression.

   c. On the Outputs Setup tab, click the  button in the *Expression/Signal/File* field.

      The new expression appears in the *Expression/Signal/File* field.

See also:

- Re-evaluating Expressions and Specifications on page 650

Loading an OCEAN or a MATLAB Measurement

Note: See `axlOutputResult` in the *Virtuoso Analog Design Environment XL SKILL Functions Reference*. See the *OCEAN Reference* for information about OCEAN script commands.

To load an OCEAN script file containing one or more output measurements or a MATLAB measurement defined in a script file (one value per script) for a test, do the following:

1. On the Outputs Setup tab, click the button.
2. In the drop-down list, select a test and choose *OCEAN script* or *MATLAB script*.

   A new row is added for the test with the output type *ocean* or *matlab*.

Tip

You can double-click on the test name or output type to select a different test or output type from the drop-down list that appears in the *Test* or *Type* column.

3. (Optional) In the *Name* field, type a name to represent the measure.
4. Double-click on the *Expression/Signal/File* field.
5. Type the name of (and location/path to) the script file.

Tip

Alternatively, you can click the browse button to open a browser window so that you can browse to locate a file. After you select a file and click *Open*, the relative path to the file appears in the *Expression/Signal/File* field.

Note: Once the script file name appears in this field, you can edit it by clicking the Edit File button E.

See also

- Editing an OCEAN or a MATLAB Script File on page 393
- Writing a MATLAB Measure on page 394
- Specifying Whether a Result Will Be Saved or Plotted on page 375
- Setting Plotting Options for Specific Tests on page 620

Editing an OCEAN or a MATLAB Script File

To edit an OCEAN or a MATLAB measurement in a script file, do the following:

1. On the Outputs Setup tab, double-click on the OCEAN or MATLAB script file you want to edit in the *Expression/Signal/File* field.
2. Click clicking the *Edit File* button.

   The script file appears in an editing window.
3. Edit the script file and save your changes.

   The program uses the new script the next time you run the test.

Writing a MATLAB Measure

When writing a MATLAB measure, you create a script in a `.m` file. You can use additional tools such as the `cds_srr` function provided in the Spectre/RF MATLAB Toolbox[1] to read the results data. You can also use the `axlCurrentResultsPath` special function in your MATLAB script file to determine the current ADE XL results path. If you want your script to write results to the Results tab of the Outputs pane, you must assign the value to the `axlResult` variable. You can plot your results in MATLAB if you include a MATLAB `plot` command in your script file.

To write a MATLAB measure for a test, do the following:

1. Use a text editor to create a `.m` file.
2. (Optional) Use the `cds_srr` function to access results data as follows:

   `cds_srr( 'path_to_psf_dir', 'analysisName', 'outputVariable' )`

   where

   `path_to_psf_dir` is the path to the `psf` directory; for example:

   `simulation/ampTest/schematic/psf`

   Tip

   You can use the `axlCurrentResultsPath` function (instead of `'path_to_psf_dir'`) as follows to return the current ADE XL results path:

   `cds_srr( axlCurrentResultsPath, … )`

   `analysisName` is the analysis name string that corresponds to the `psf` subdirectory that contains the results; for example:

   `tran-tran`

   `outputVariable` is the output variable name

   For example:

   `cds_srr( 'simulation/ampTest/schematic/psf', 'tran-tran', 'out' )`

3. If you want your script to write results to the Results tab of the Outputs pane, you must assign the value to the `axlResult` variable as follows:

   `axlResult = resultStatement`

1. Installed in your MMSIM 6.x hierarchy under `tools/spectre/matlab`.

For example:

```
axlResult = max( out.time )
```

4. Save your changes and load this `.m` file.

Important

Matlab prefers current directory over any other path. So, if the Matlab script file (.m file) exists in the current directory, it is always run from there. To run a matlab script saved at a location other than the current run directory, delete the script from the current directory.

Example scripts

The following script creates a MATLAB plot using the specified X and Y values.

```
figure
x = [1 2 3]
y = [2 0 6]
plot(x,y)
```

The following script accesses the value of the `out` output variable from the current transient results directory using `axlCurrentResultsPath`, plots `out` using MATLAB (Voltage on the X axis and time on the Y axis), and writes the measured value (for total simulation time) to the Results tab of the Outputs pane.

```
out = cds_srr( axlCurrentResultsPath,'tran-tran', 'out' )
fig = figure
plot( out.V,out.time )
axlResult = max( out.time )
```

Copying Outputs

You can copy outputs within a test or from one test to another. You can also copy and paste the contents of an output. This allows you to quickly setup outputs for your tests.

For more information, see the following topics:

- Copying Outputs Within and Across Tests on page 396.
- Copying the Contents of an Output on page 397

Copying Outputs Within and Across Tests

You can copy outputs within a test or from one test to another. For example, if you have specified an output expression for one test, you can do the following:

- Make a copy of the output expression within the same test.
- Copy the output expression to another test.

Important

An operating region specification is copied from one test to another test only if the design and testbench of both the tests is same.

To copy outputs within and across tests, do the following:

➤ On the Outputs Setup tab of the Outputs pane, right-click the row for the output you want to copy, select *Copy To Test*, then do one of the following:

- ❑ To copy the output within a test, select the name of the same test.
- ❑ To copy the output to another test, select the name of that test.
- ❑ To copy the output to all tests, select *All*.

To copy multiple outputs, hold down the *Shift* key (for contiguous selection) or the *Ctrl* key (for noncontiguous selection) and click the output to add it to the selection set. Right-click and choose *Copy To Test*, then select the name of the target test or *All*.

Note: An output will be copied across tests only if an output with the same name does not exist in the target test. For example, if you are copying an output expression named `DCGain` from one test to another, and the target test has an output with the same name, the output expression will not be copied.

Copying the Contents of an Output

To copy the contents (name, expression, file name, specification, weight, units or digits) of an output, do the following:

1. On the row for the output in the Outputs Setup tab of the Outputs pane, select the content you want to copy.

   For example, to copy an expression, select the expression in the *Expression/Signal/File* column.

2. Press *Ctrl+C* to copy the content.
3. Select the cell where you want to paste the content.

   For example, to paste an expression, select a cell in the *Expression/Signal/File* column.

4. Press *Ctrl+V* to paste the content.

Adding User-Defined Columns in the Outputs Setup Tab

In addition to the default set of columns that appear in the *Outputs Setup* tab, you can add one or more user-defined or custom columns. These user-defined columns can be used to save additional information, such as comments, target values for specifications, or other notes related to a measurement.

To help you use the information provided in the user-defined columns while analyzing results, ADE XL makes this information available in the following ways:

- User-defined columns are displayed on the *Results* tab. These columns are visible in the `Detail`, `Optimization`, `Summary`, and `Yield` result views.

  Note: Currently, these columns are not available in the `Detail-Transpose` view.

- If you add a new column and re-evaluate results, the new column is displayed on the *Results* tab.
- When you export the outputs to CSV or HTML files, the user-defined columns are also saved with other columns.
- These columns are displayed in the Specification summary sheets and datasheets.

For more information, see the following topics:

- Adding a User-Defined Column
- Renaming a User-Defined Column
- Deleting a User-Defined Column
- Hiding a User-Defined Column

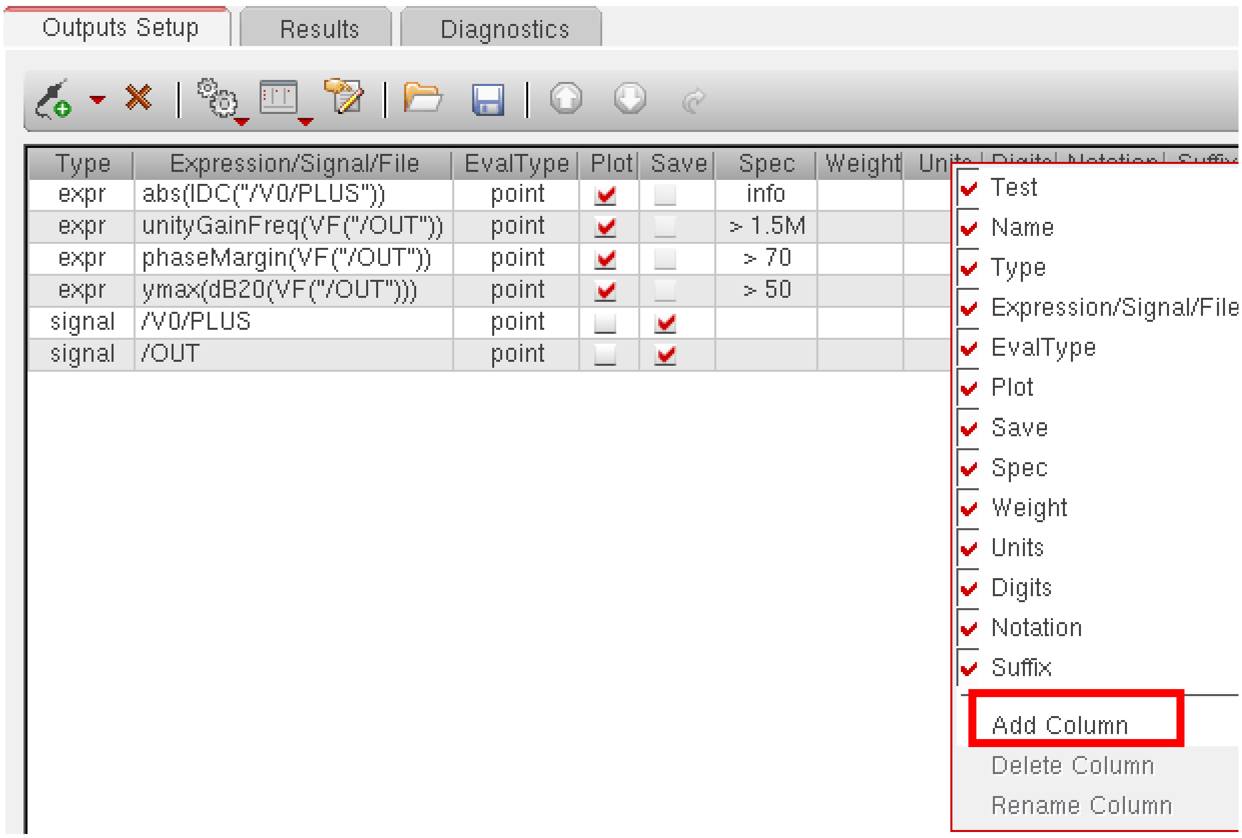
Tip

To know more about the SKILL APIs required to add a new user-defined column or to get or set value in a SKILL script, refer to SKILL Functions for Outputs in the *ADE XL SKILL Reference Guide*.

Adding a User-Defined Column

To add a user-defined column in the *Outputs Setup* tab, do the following:

1. Right-click on the title of any existing column and choose *Add Column* from the context-sensitive menu that appears.

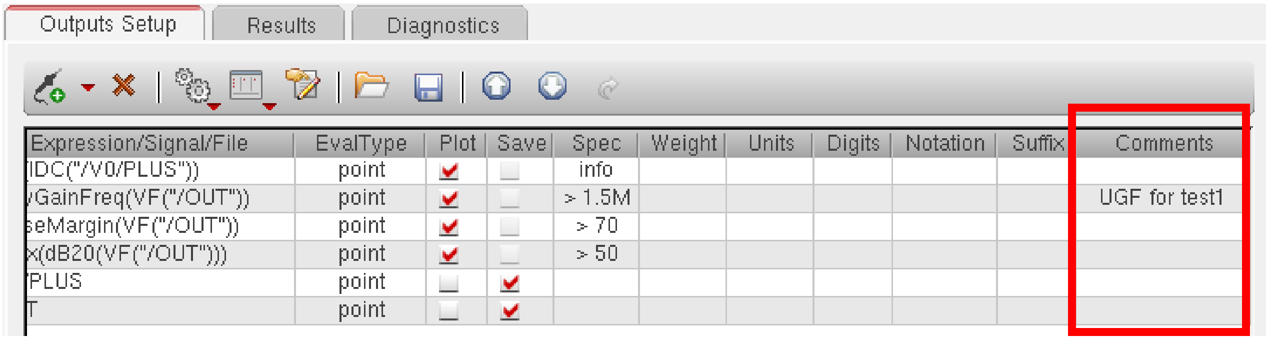


The Add Column form is displayed.

2. In the *New Column Name* field, specify a name for the new column to be added.

3. Click *OK*.

A new column is added to the right of the existing columns.



You can add any information in a custom column. The information in this column is retained and displayed on the *Results* tab post simulation run.

Also see:

- Renaming a User-Defined Column
- Deleting a User-Defined Column
- Hiding a User-Defined Column

Renaming a User-Defined Column

If required, you can rename a user-defined column. For this, do the following:

➡ Right-click on the column in the *Outputs Setup* tab and choose *Rename Column* from the context-sensitive menu.

The Rename Column form is displayed.

1. In the *Rename Column to* field, specify a new name for the column.
2. Click *OK*.

Deleting a User-Defined Column

To delete a user-defined column, do the following:

➡ Right-click on the column and choose *Delete Column*.

The column is removed from the *Outputs Setup* tab.

Important

You can only delete a custom or user-defined column. It is not possible to delete a standard column from the *Outputs Setup* tab. However, you can hide or show selected columns, as required.

Hiding a User-Defined Column

To hide a user-defined column on the *Outputs Setup* tab, do the following:

➡ Click (*Configure what is shown in the table*) on this tab and clear *User-Defined Columns*.

To hide a user-defined column on the Results tab, do the following:

➡ Click (*Configure what is shown in the table*) on this tab and clear *User-Defined Columns*.

All the custom columns are hidden.

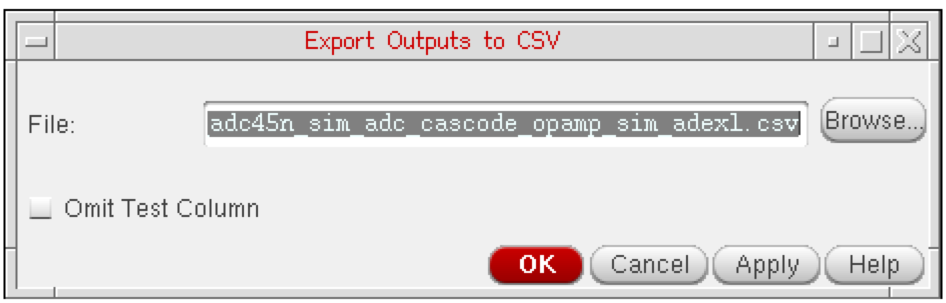
Exporting Outputs to a CSV File

You can save the outputs from the *Outputs* tab to a CSV file. This helps in saving outputs in a format from which they can be later reused and imported in another adexl view.

To export outputs to a CSV file, do the following:

1. Ensure that all the required outputs are defined on the *Outputs Setup* tab.
2. Click *Export Outputs to CSV File* on the toolbar on the *Outputs Setup* tab.

The **Export Outputs to CSV** form is displayed, as shown below.

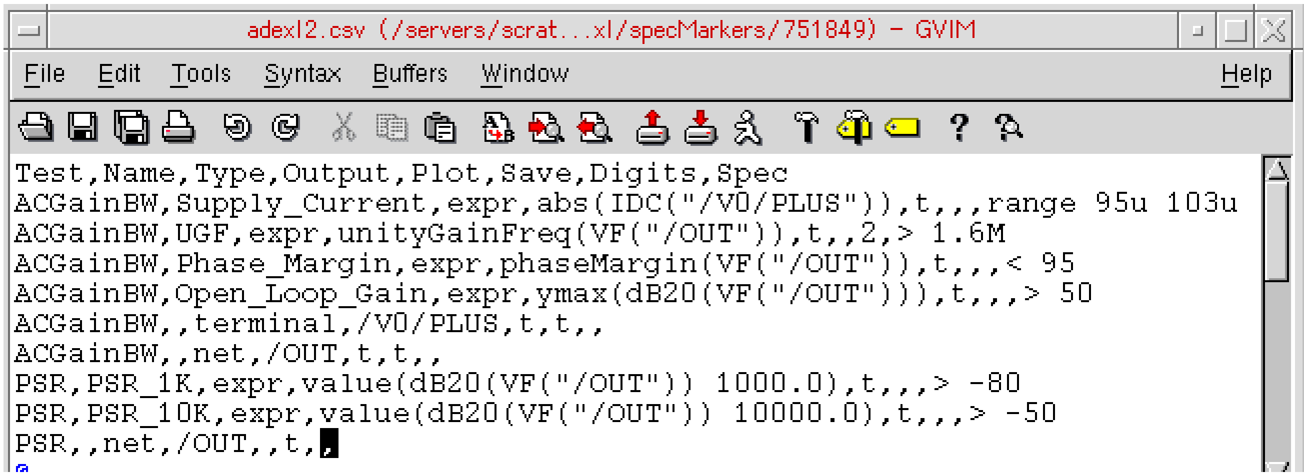


3. (Optional) Change the file name in the *File* field. By default, ADE XL uses the name of the adexl view as the name of the CSV file.
4. (Optional) Click *Browse* to choose a directory where you want to save the CSV file. By default, the file is saved in the current working directory.
5. (Optional) Select the *Omit Test Column* check box to specify that the test names are not to be saved in the CSV file.

   By default, ADE XL saves the test name along with each saved output to indicate the test to which this belongs. Next time, when you import the outputs from the CVS file, the outputs will be assigned to the given test name only.

   When the outputs are saved without the test name, you can load them later for another test as well. For more details, refer to Importing Outputs from a CSV File.
6. Click *Apply* and then *OK* to export the outputs and to close the form.

All the outputs defined on the *Outputs* tab are saved in the specified CSV file, as shown below.




```
Test,Name,Type,Output,Plot,Save,Digits,Spec
ACGainBW,Supply_Current,expr,abs(IDC("/V0/PLUS")),t,,,range 95u 103u
ACGainBW,UGF,expr,unityGainFreq(VF("/OUT")),t,,2,> 1.6M
ACGainBW,Phase_Margin,expr,phaseMargin(VF("/OUT")),t,,,< 95
ACGainBW,Open_Loop_Gain,expr,ymax(dB20(VF("/OUT"))),t,,,> 50
ACGainBW,,terminal,/V0/PLUS,t,t,,
ACGainBW,,net,/OUT,t,t,,
PSR,PSR_1K,expr,value(dB20(VF("/OUT")) 1000.0),t,,,> -80
PSR,PSR_10K,expr,value(dB20(VF("/OUT")) 10000.0),t,,,> -50
PSR,,net,/OUT,,t,,
```

Important Points to Note:

- In the saved CSV file, the *Test* column has been saved to indicate the test for which an output is saved. If you omit to save the test column, this information is not saved in the CSV file.
- An appropriate output type, such as, `net`, `terminal`, `expr`, or `corners` is used to indicate the output type. `corners` is the output type used for the expressions measured across corners.
- If no value is specified for any column, it is considered as `null`.
- It is mandatory to specify at least the *Name* column or the *Signal* and *Expression* columns in the CSV file.
- If you override an output specification for a corner, the overridden spec is also saved with the exported output. When you import such an output, the overridden spec value is applied to the same corner, if it exists in the ADE XL setup.
- Outputs that include the following are ignored while exporting outputs to CSV:
  - OCEAN measures
  - MATLAB scripts
  - device checks
  - area specifications

- operating region specifications

- By default, only the Test, Name, Type, Output, Plot, Save, and Spec columns are saved. However, if you have specified values in other columns as well, all other columns, such as Notation, are also saved in the CSV file.
- In the earlier releases, outputs exported from ADE L were saved in the `.txt` format. Now, the outputs from ADE L are also saved in the CSV format by default. However, you still have an option to save them in the `.txt` format as before. For more details, refer to Importing and Exporting Outputs in ADE L Environment in the Analog Design Environment L user guide.
- Outputs that were earlier exported from the ADE L environment in a `.txt` file can be imported in ADE XL.

Related topics:

- Importing Outputs from a CSV File on page 404
- Related SKILL functions

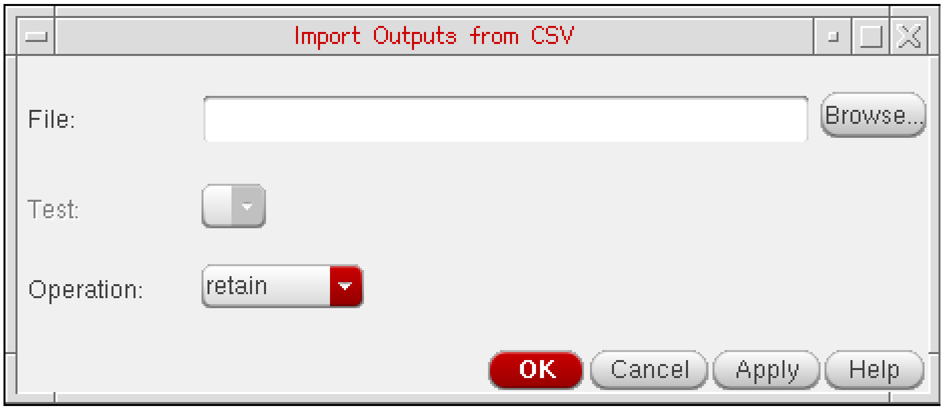
Importing Outputs from a CSV File

If you have outputs exported and saved in a CSV file, you can import those to the same or a different adexl view, as required.

To import the outputs from a CSV file, do the following:

1. Open the adexl view to which you need to import the outputs.
2. Click *Import Outputs from CSV File* on the toolbar on the *Outputs Setup* tab.

The **Import Outputs from CSV** form is displayed, as shown below.



3. Click *Browse* and select the CSV file from which you need to import the output details.

   ADE XL reads the specified file. If the details in the file were saved with the test column, the *Test* drop-down list on this form remains disabled.

   If the selected CSV file does not contain the details of the test column, the *Test* drop-down list is enabled and the names of all the tests in the current adexl view are listed in it.

4. (Optional) If the *Test* drop-down list is enabled, select the name of the test for which you wish to import the outputs from the specified CSV file.

   Select `Import to All Tests` to import the complete set of outputs for all the tests in the current adexl view.

5. Select an appropriate option from the *Operation* field to specify how you want to use the output rows that already exist before you import details from the CSV file.

   Following are the three possible operations to choose from:

   - `retain`: If an output already exists in the Outputs Setup tab with a name which is same as that of an output being imported, the import is not done and the existing output is retained.

     In this case, a message similar to the one shown below is displayed in CIW:

     ```
     *Info*    Output "Supply_Current" exists in the current setup and the
     operation is "retain". Skipping import of row 2 from CSV file "./out2.csv".
     ```

     An existing output which has the same signal name as that of an output being imported is also retained.

   - `merge`: If an existing output for a given test has the same name as that of an output being imported, it is deleted and is overwritten by the imported output.

     In this case, a message similar to the one shown below is displayed in CIW:

```
*Info*    Output "CMRR@10M" exists in the current setup and the operation
is "merge". Deleting this output from the current setup and importing it
from CSV file "./out1.csv".
```

Note: Other outputs for which the names do not match with the outputs being imported are also retained.

- ❑ `overwrite`: All existing outputs for the given test are deleted and all the imported outputs are used.

  In this case, a message similar to the one shown below is displayed in CIW:

```
*Info*    The import outputs from CSV file operation is "overwrite". The
original outputs in test "Offset" are deleted.
```

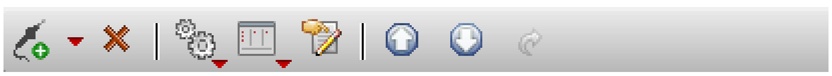
6. Click *Apply* and then *OK* to import the outputs and to close the form.

Related topics:

- Exporting Outputs to a CSV File on page 402
- Related SKILL functions

Configuring How Outputs Appear on the Outputs Setup Tab

By default, the Outputs Setup tab of the Outputs pane displays the outputs added for all the tests. You can change the order in which outputs or columns are displayed or you can show or hide the outputs or columns by using the toolbar on top of this tab.



For more information about viewing outputs in the Outputs Setup tab, see the following topics:

- Changing the Order of Outputs on page 407
- Sorting the Outputs on page 408
- Hiding and Showing Outputs on page 408
- Hiding and Showing Output Details on page 409
- Changing the Order of Columns on page 409
- Hiding and Showing Columns on page 409

Changing the Order of Outputs

To change the order of outputs, do one of the following:

- Click on the row for an output and:
  - Click the  button to move it one level up.
  - Click the  button to move it one level down.
- Drag and drop the output at the desired location.

When you run a simulation, the outputs for a test are displayed in the same order in the Results tab of the Outputs pane.

Note: Order of output rows is overridden if you sort the outputs by a column. For details, see Sorting the Outputs.

Sorting the Outputs

By default, outputs are sorted by test names. However, you can sort them by any column on this tab. For example, you can sort the outputs by output name or by output type.

To change the sorting order by a column, click on the header of that column. The up or down arrow on the top-right area of the column header shows if the rows are being arranged in the ascending or the descending order. All the output rows are sorted according the values in the selected column.

Important

Before sorting the rows, if you had rearranged the output rows in a specific order by using the or buttons, those settings are overridden after you sort the outputs by a column. However, you can restore the same arrangement by clicking on the (*Reset table to display per-test evaluation order)* button on the Outputs Setup tab. The tool groups the outputs by test name and within each test, displays the outputs in the user-specified order.

Hiding and Showing Outputs

To view only the outputs added for a specific test, do the following:

1. Click the button on the Outputs Setup tab and select *Hide All Tests*.
2. Click the button on the Outputs Setup tab and select a specific test.

To view the outputs added for all the tests, do the following:

➤ Click the button on the Outputs Setup tab and select *Show All Tests*.

To view only the outputs added the tests that are enabled in the Data View pane, do the following:

➤ Click the button on the Outputs Setup tab and select *Show Enabled Tests*.

To view only the outputs added the tests that are disabled in the Data View pane, do the following:

➤ Click the button on the Outputs Setup tab and select *Show Disabled Tests*.

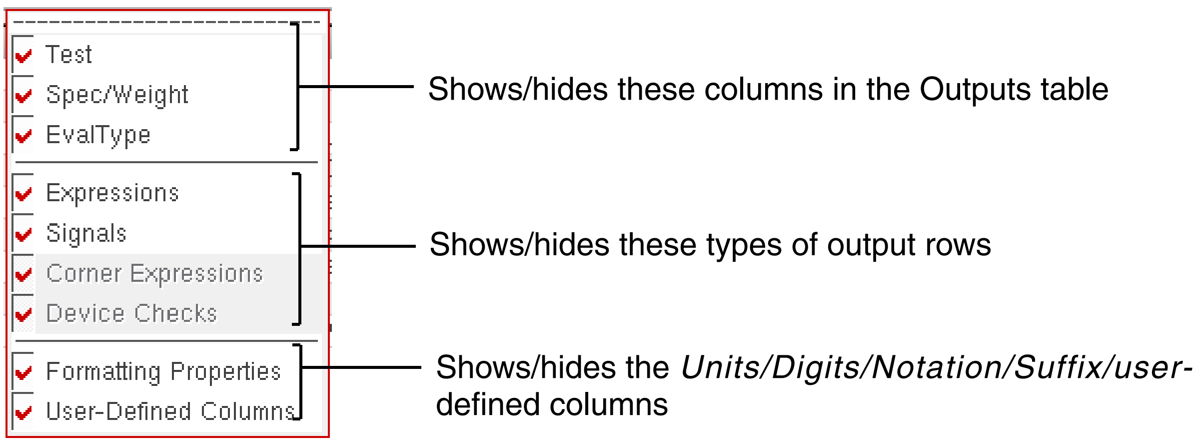
To hide the outputs added for all tests, do the following:

➤ Click the button on the Outputs Setup tab and select *Hide All Tests*.

Hiding and Showing Output Details

To show or hide output details in the Outputs table, do the following:

1. Click the *Configure what is shown in the table* ( ) button on the Outputs Setup tab. The following drop-down list appears.



2. Select/deselect the check box next to an item in this list to show or hide the related information in the Outputs table.

Changing the Order of Columns

To change the order of columns in the Outputs Setup tab of the Outputs pane, do the following:

1. Click on a column name.
2. Drag and drop the column at the desired location.

Hiding and Showing Columns

To hide the columns in the Outputs Setup tab of the Outputs pane, do the following:

Right-click on a column name to display a pop-up menu listing all the columns. A check mark next to a column name indicates that the column is being displayed.

➡ To hide a column, click the column name.

➡ To unhide the column, right-click on a column name and click the column once again in the pop-up menu.

Selecting Outputs to Save or Plot

You can select outputs (signals and expressions) to save or plot by selecting the *Plot* or *Save* check boxes on the Outputs Setup tab of the Outputs pane.

To select an output for plotting, do the following:

➤ In the *Plot* column, select the *Plot* check box.

The program plots the item when the simulation finishes.

Tip

To select more than one output for plotting, select the outputs, right-click and choose *Enable Plot* from the pop-up menu.

To select an output for saving, do the following:

➤ In the *Save* column, select the *Save* check box.

The program saves the item to the results database.

Tip

To select more than one output for saving, select the outputs, right-click and choose *Enable Save* from the pop-up menu.

See also "Specifying Whether a Result Will Be Saved or Plotted" on page 375.

Removing Outputs

You can remove outputs (signals and expressions) from the save or plot set by deselecting the *Plot* or *Save* check boxes on the Outputs Setup tab of the Outputs pane. You can also remove outputs altogether from the test setup.

To remove an output from the plot set, do the following:

➤ In the *Plot* column, deselect the *Plot* check box.

The program does not plot the item when the simulation finishes.

Tip

To remove more than one output from the plot set, select the outputs, right-click and choose *Disable Plot* from the pop-up menu.

To remove an output from the save set, do the following:

➤ In the *Save* column, deselect the *Save* check box.

The program does not save the item to the results database.

Tip

To remove more than one output from the save set, select the outputs, right-click and choose *Disable Save* from the pop-up menu.

To remove an output from the test setup, do the following:

➤ In the Outputs Setup tab, select an output, right-click and choose *Delete Output*.

The program removes the item from the test setup.

Tip

To remove more than one output from the test setup, select the outputs, right-click and choose *Delete Output*.

See also "Specifying Whether a Result Will Be Saved or Plotted" on page 375.

Saving All Voltages or Currents

To save all of the node voltages and terminal currents, do the following:

1. On the Outputs Setup tab of the Outputs pane, right-click a test and choose *Save All*.

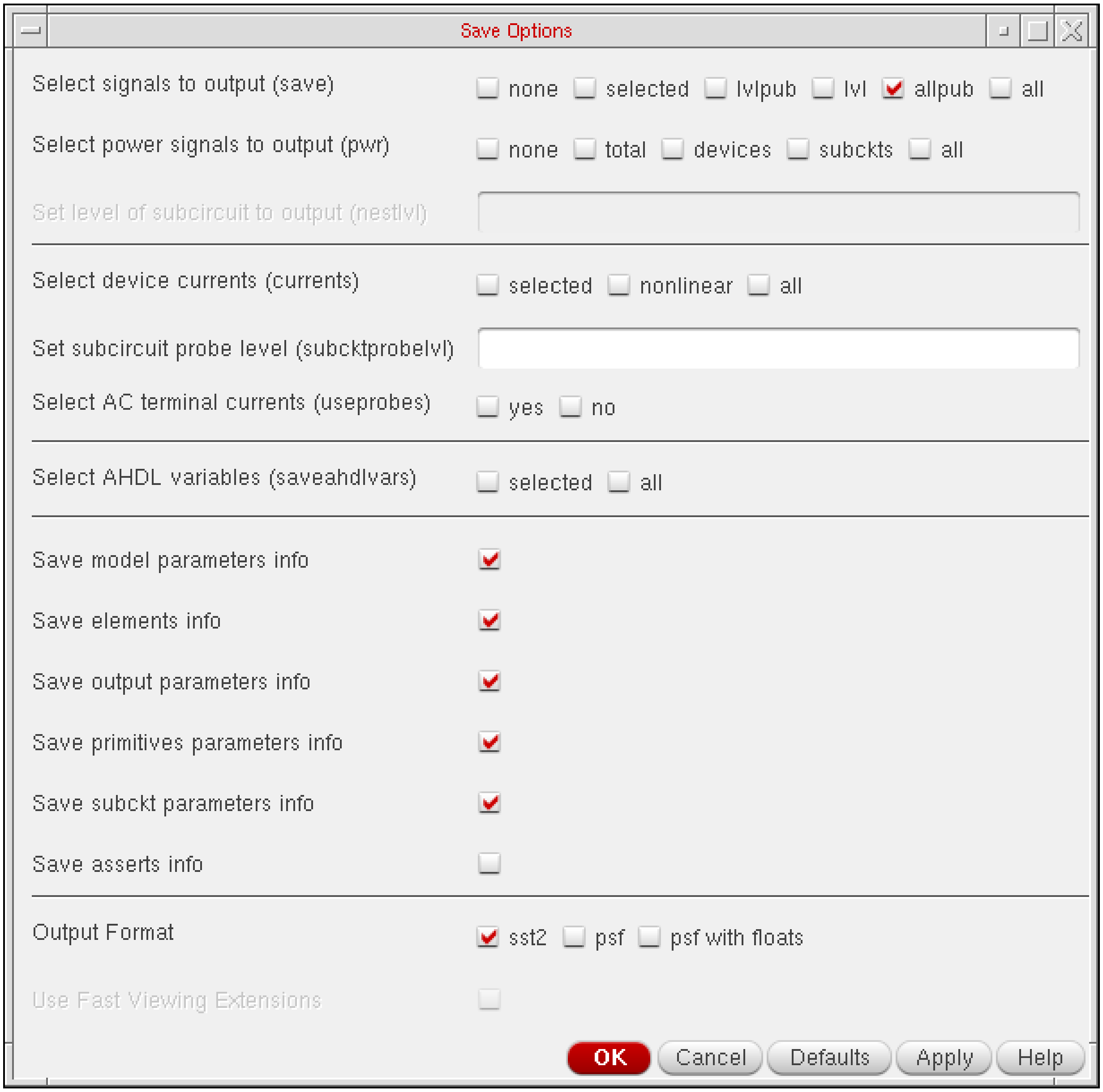
   The Save Options or Keep Options form appears, depending on your target simulator. The options that appear on this form vary according to your target simulator:

   - Save Options form for Spectre Simulations on page 413
   - Keep Options form for UltraSim Simulations on page 414
   - Save Options form for AMS Simulations on page 415
   - Save Options form for SpectreVerilog Simulations on page 416
   - Keep Options form for UltraSimVerilog Simulations on page 417
   - Keep Options form for HspiceD Simulations on page 419

2. Select the values you want to save and click *OK*.

Save Options form for Spectre Simulations

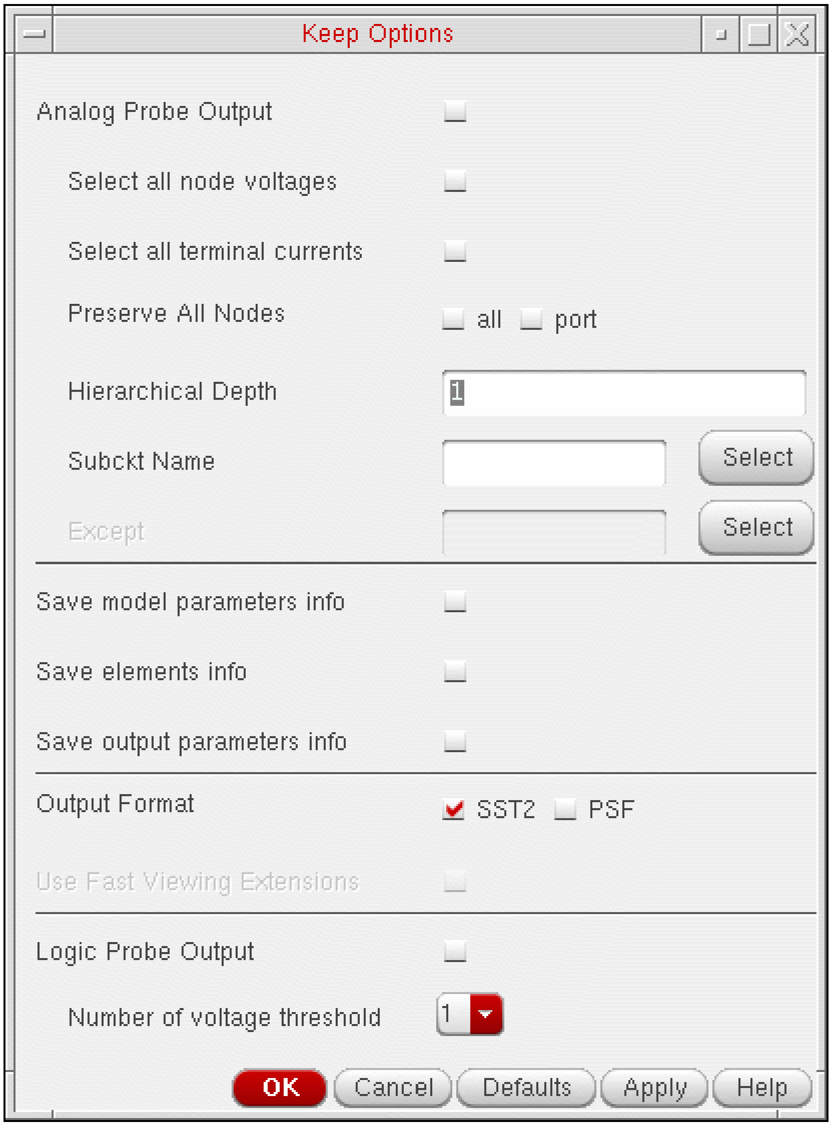
For Spectre simulations, the following Save Options form appears:



For more information, see "Saving All Voltages or Currents" in the *Virtuoso Analog Design Environment L User Guide*.

Keep Options form for UltraSim Simulations

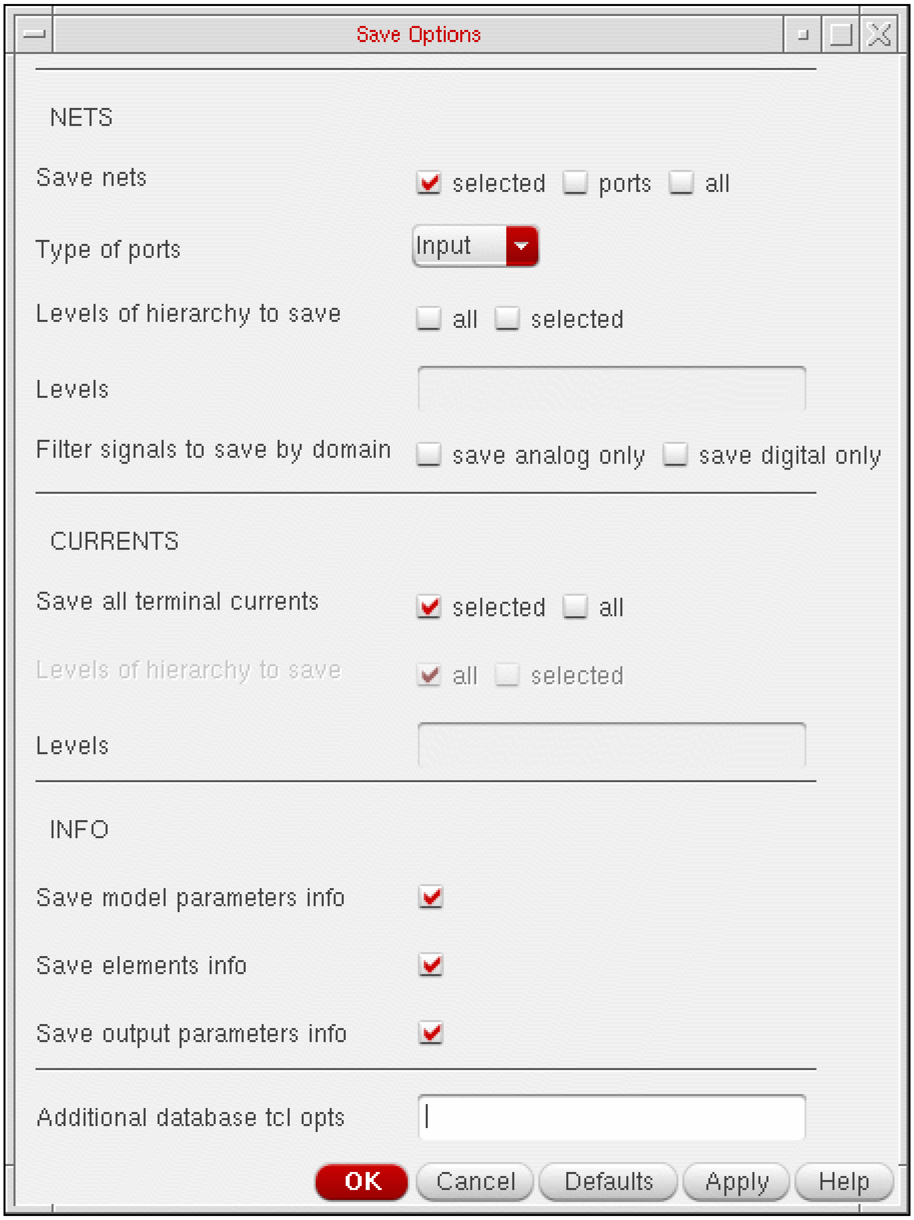
For UltraSim Simulations, the following form appears:



For more information, see ".probe" and "save" in the *Virtuoso UltraSim Simulator User Guide*.

Save Options form for AMS Simulations

For AMS Simulations, the following form appears:



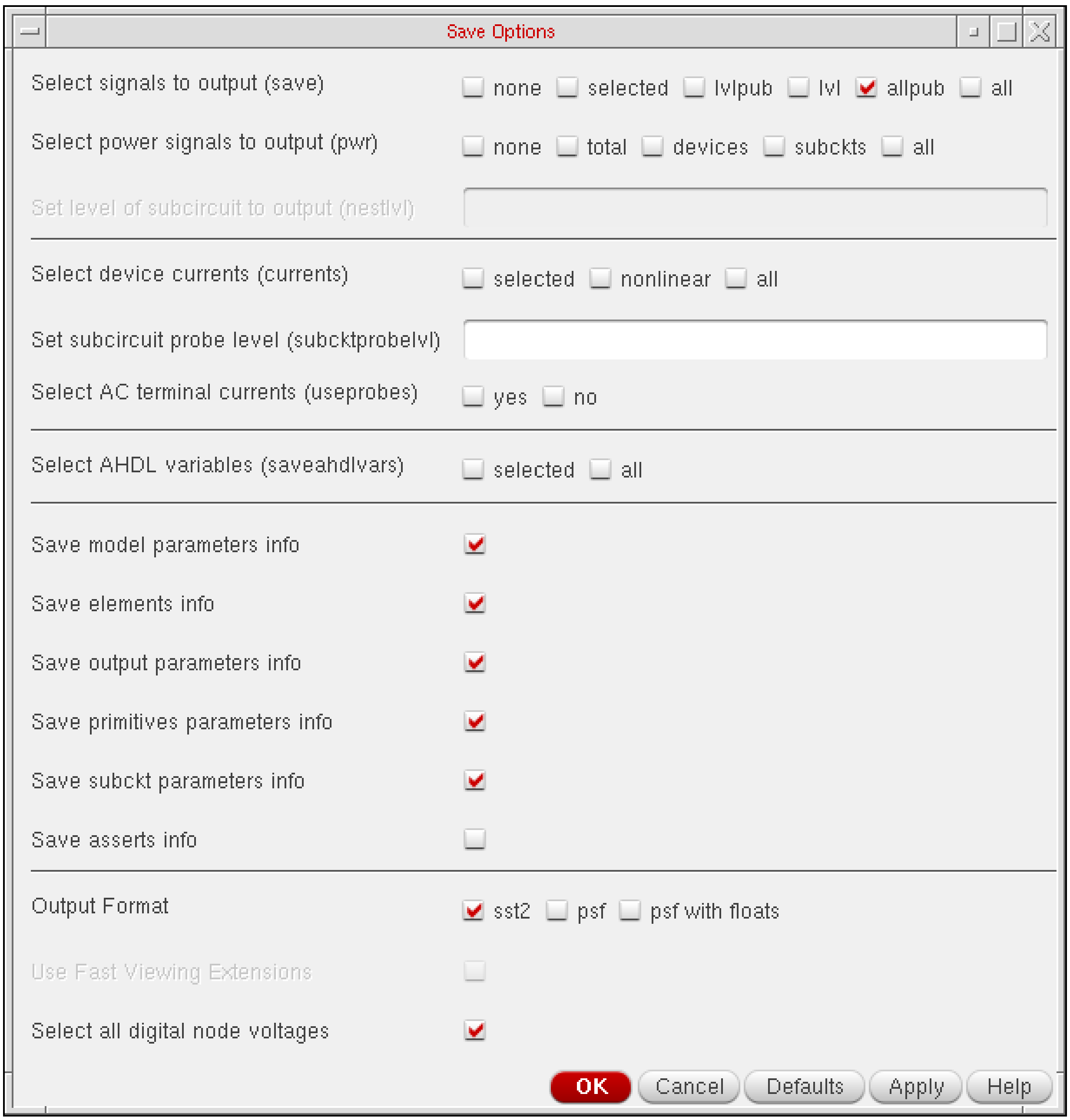
For more information, see Displaying and Saving Information (info) in the *Virtuoso AMS Designer Simulator User Guide*.

Save Options form for SpectreVerilog Simulations

Important

The SpectreVerilog circuit simulator is available only in IC6.1.6 release.

For SpectreVerilog Simulations, the following form appears:



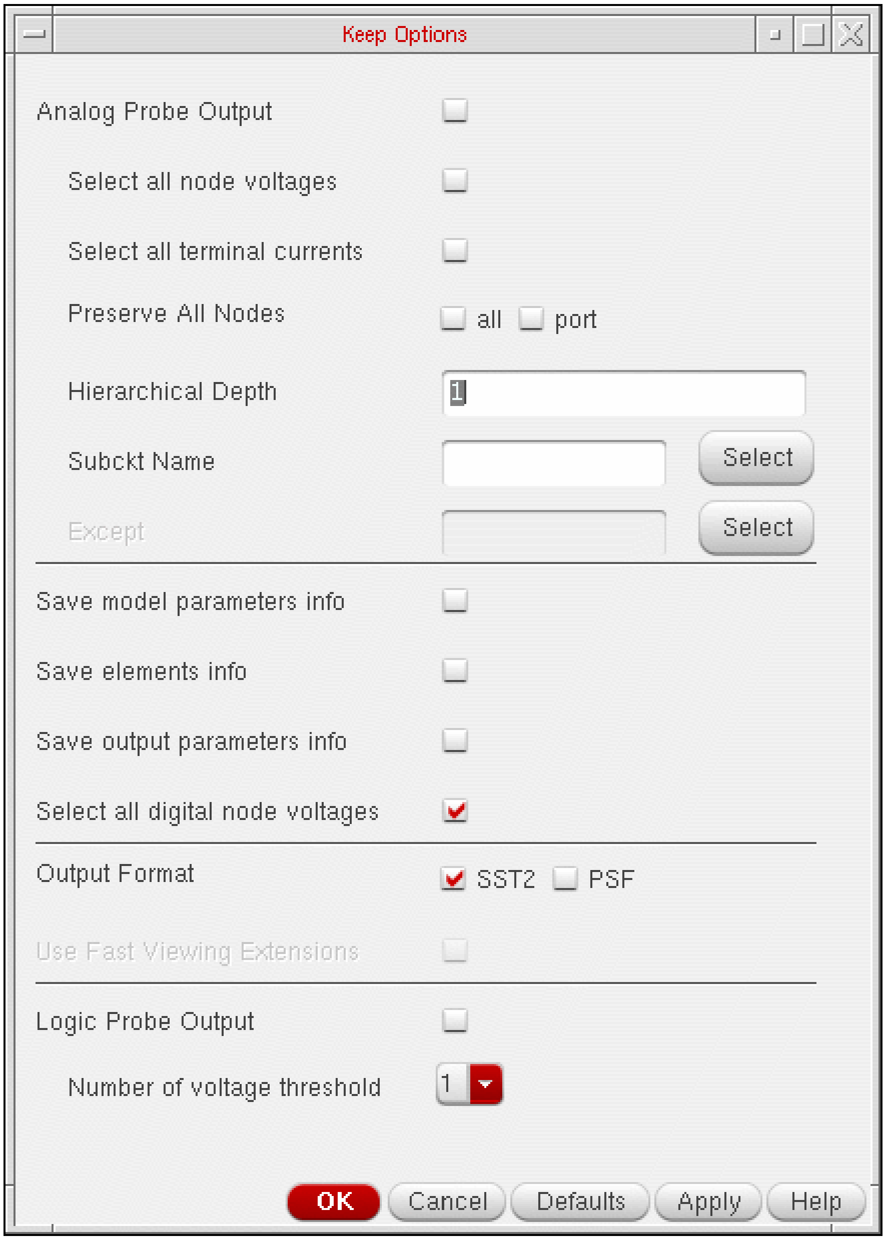
For more information, see "Saving All Voltages or Currents" in the *Virtuoso Analog Design Environment L User Guide*.

Keep Options form for UltraSimVerilog Simulations

Important

The UltrasimVerilog circuit simulator is available only in IC6.1.6 release.

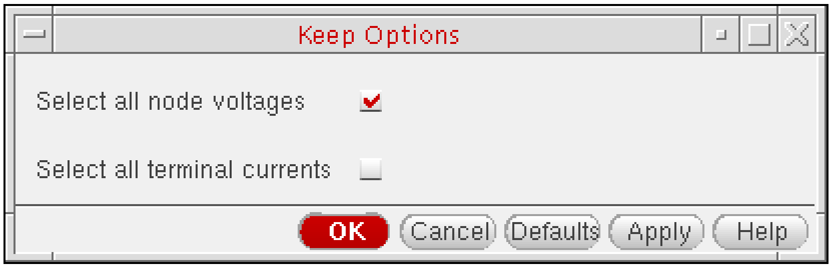
For UltraSimVerilog simulations, the following form appears:



For more information, see ".probe" and "save" in the *Virtuoso UltraSim Simulator User Guide*.

Keep Options form for HspiceD Simulations

For HspiceD Simulations, the following form appears:



12

Device Checking

You can use Spectre's device checking feature to determine whether elements in your circuit are violating predefined safe operating areas. You can specify rules that check operating point parameters, as well as expressions that combine these parameters. If your circuit has any violations, you can highlight violating devices on the schematic, view the details of violations for a device check, and print a summary of all the violations.

See the following topics for more information:

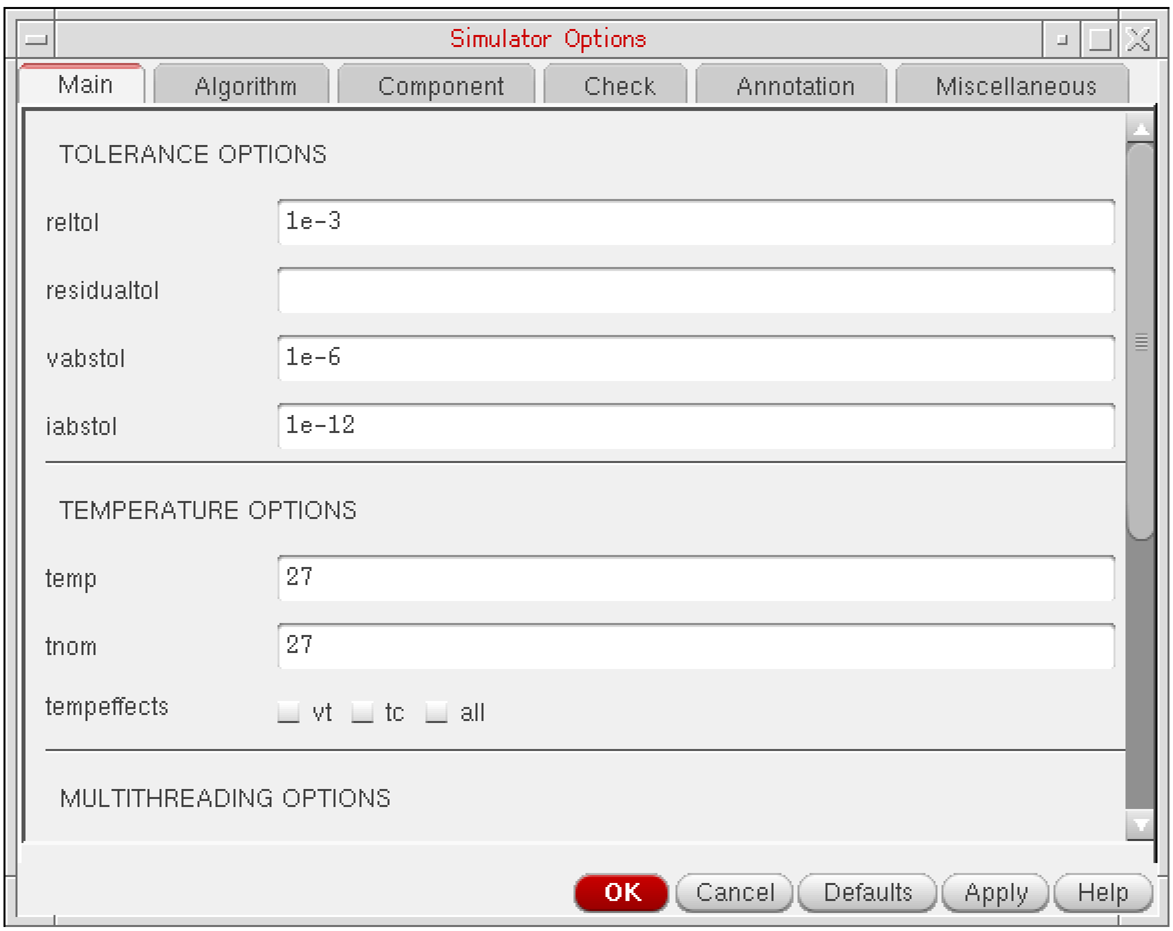
Enabling and Disabling Device Checking

You can enable and disable device checking for individual tests.

To enable device checking for a test, do the following:

1. On the Data View assistant pane, right-click the test and choose *Options – Analog*.

   The Simulator Options form appears.



2. Click the *Check* tab.

3. In the *DEVICE CHECKING OPTIONS* section, select the *yes* check box next to *dochecklimit*.

4. Click *OK*.

   The program will check for violations when you run the simulation.

To disable device checking for a test, do the following

1. On the Data View assistant pane, right-click the test and choose *Options – Analog*.

   The Simulator Options form appears.

2. Click the *Check* tab page.
3. In the *DEVICE CHECKING OPTIONS* section, select the *no* check box next to *docheck limit*.
4. Click *OK*.

See also:

Specifying Options for Writing Violations Information

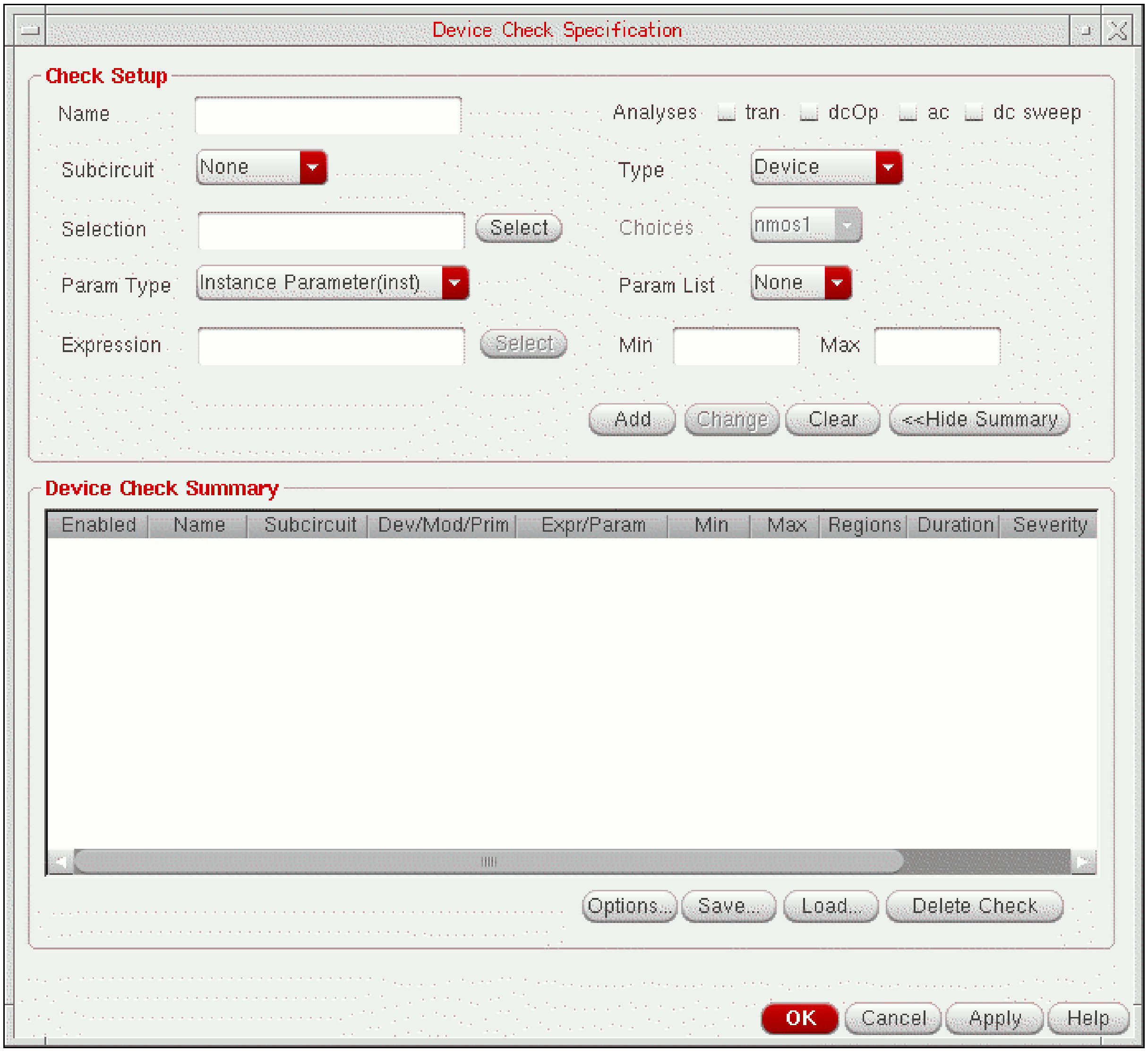
Setting Up Device Checks

You can set up device checks for individual tests.

To set up device checks for a test, do the following:

1. On the Outputs Setup tab, click the  button.
2. In the drop-down list, select a test and choose *Device Check*.

   The Device Check Specifications form appears.



From this form, you can perform the following tasks:

- ❑ Add or change device checks
- ❑ Set global device check options

The device checks that are enabled in the Device Check Specification form are displayed in the Outputs Setup tab of the Outputs pane.

- The device check name is displayed in the *Name* field.
- The type *check* in the *Type* column indicates that it is a device check.

- The *Expression/Signal/File* column displays:
  - The text *parameter,* if the device check is of type Device, Model, Primitive or Parameter.
  - The expression, if the device check is of type Expression.

For more information, see the following tasks:

- Adding or changing device checks
- Setting global device check options

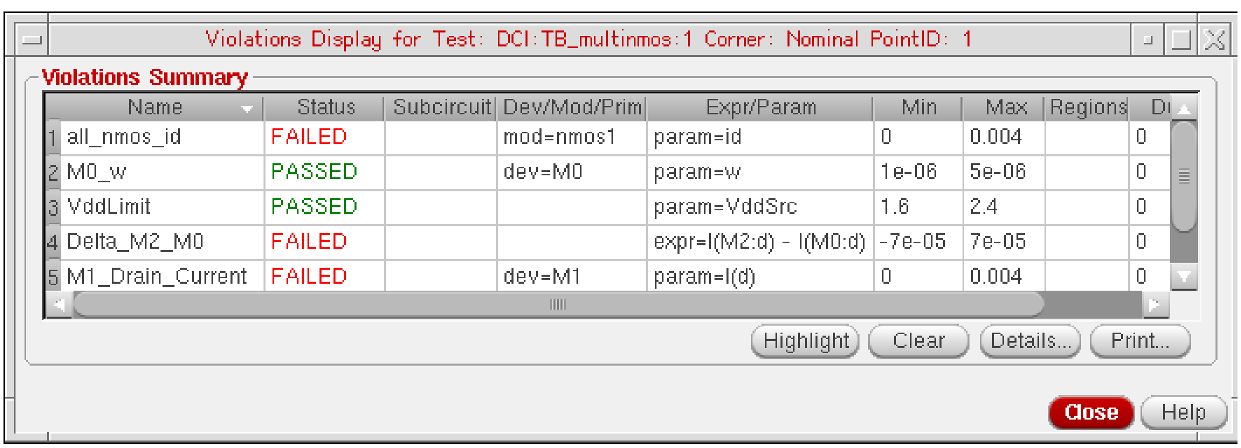
Viewing and Printing Device Check Violations

A violation occurs when the value of a device, parameter, or expression that you have defined as a device check falls outside the specified operating range.

To view all the device check violations for the nominal corner, do the following:

➤ On the Results tab of the Outputs pane, right-click a device check name in the *Output* column and choose *Violations Display – All checks for Nominal*.

The Violations Display form appears displaying the device check violations for the nominal corner.



For more information about using this form, see the following tasks:

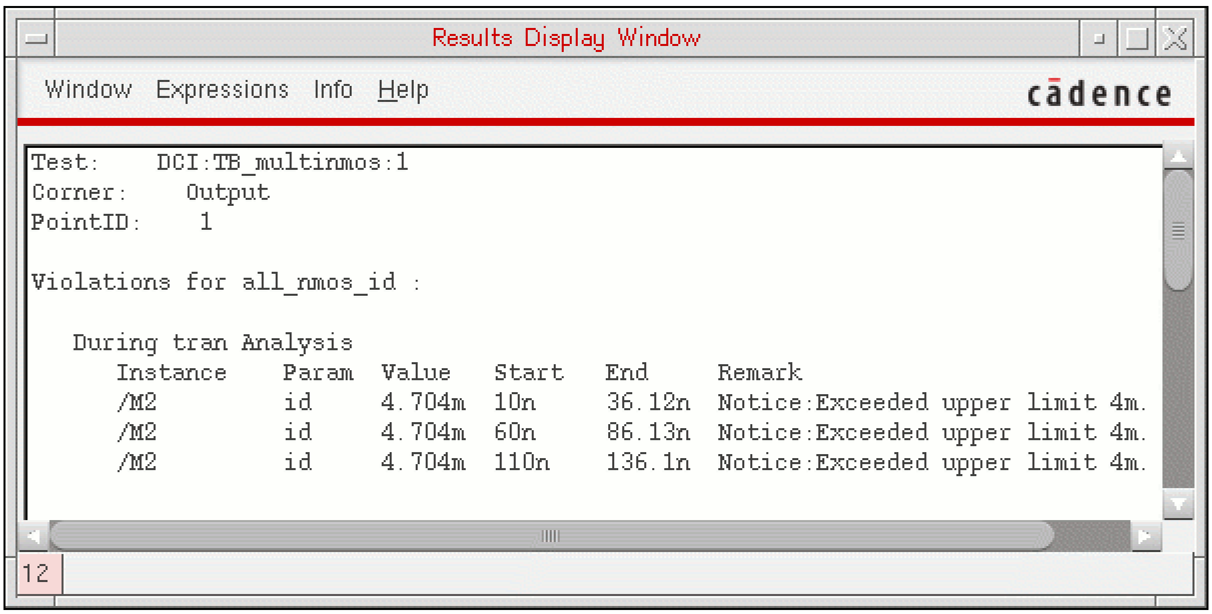
- Highlighting device check violations on the schematic
- Unhighlighting device check violations on the schematic

- ❑ Viewing the details of device check violations
- ❑ Printing device check violations

To view the violations for a specific device check at the nominal corner, do the following:

➤ On the Results tab of the Outputs pane, right-click a device check name in the *Output* column and choose *Violations Display – Selected Checks.*

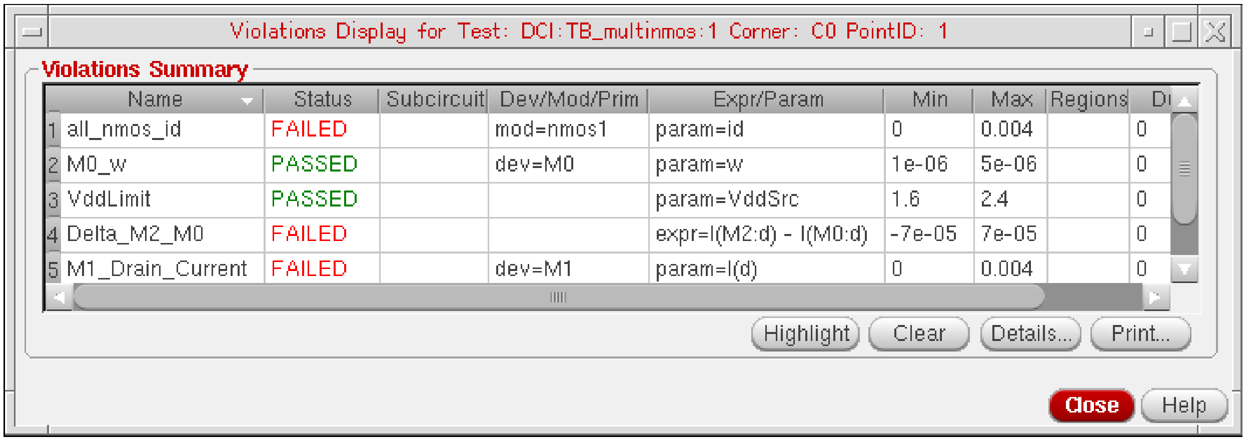
The Results Display Window form appears displaying the violations for the specific device check at the nominal corner.



To view all the device check violations for a corner, do the following:

➤ On the Results tab of the Outputs pane, right-click on the column for a corner and choose *Violations Display – All checks for* <`corner_name`>.

The Violations Display form appears displaying the device check violations for the corner.



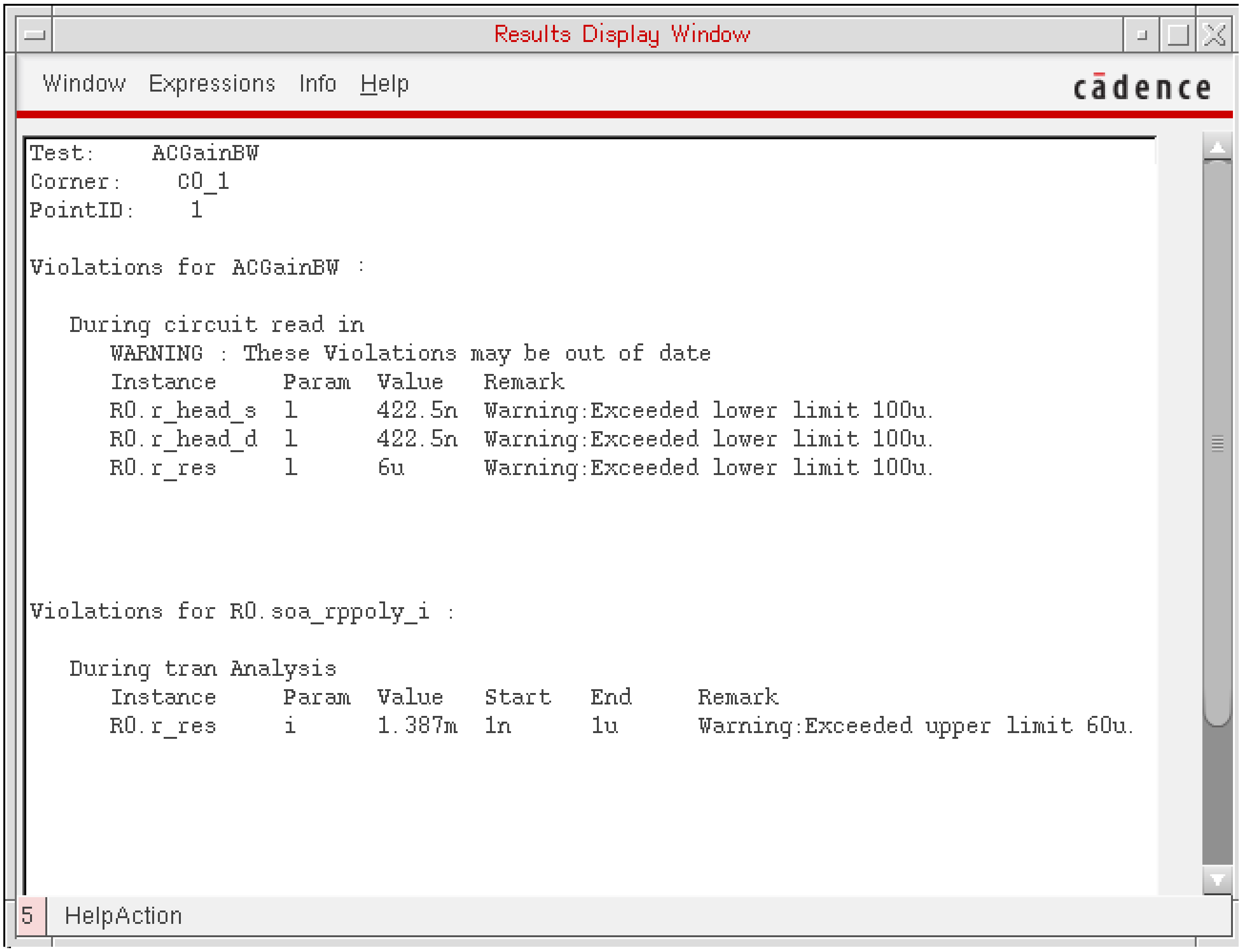
For more information about using this form, see the following tasks:

- Highlighting device check violations on the schematic
- Unhighlighting device check violations on the schematic
- Viewing the details of device check violations
- Printing device check violations

To view the violations for a specific device check at a corner, do the following:

➤ On the Results tab of the Outputs pane, right-click on the corner column in the row for the device check and choose *Violations Display – Selected Checks.*

The Results Display Window form appears displaying the violations for the specific device check at the corner.



Note that the violations for device checks (parameters and expressions) for model files are also displayed in the Results Display Window. For example, in the figure shown above, the violation for `R0.soa_rppoly_i` is for a parameter specified in the model file.

13

Running Simulations

Before you run a simulation, you should

- Start the ADE XL environment
(see “Getting Started in the Environment” on page 31)
- Specify tests and analyses
(see “Specifying Tests and Analyses” on page 79)
- Select data to save and plot
(see “Selecting Data to Save and Plot” on page 369)
- Set up job policies for local, remote, or distributed simulation
(see “Setting Up Job Policies” on page 431)
- Set up run options
(see “Setting Up Run Options” on page 451)
- Set up pre-run scripts
(see “Running Pre-run Scripts before Simulation Runs” on page 455)

See the following sections for more information:

After simulating, you can plot results (see Chapter 14, "Viewing, Plotting, and Printing Results").

Setting Up Job Policies

When you run a simulation in ADE XL, it starts an IC Remote Processes (ICRPs) where it runs simulations. Each ICRP is also called a job and can be configured to run one or more simulation points. ADE XL internally uses these ICRPs/jobs to efficiently distribute time-consuming tasks that can be performed in parallel. Settings for these ICRPs such as how many remote processes to start; where the processes should run, on local or remote machines; or the time for which a remote process should stay active and wait for a simulation to run; are set as a job policy.

The default job policy for ADE XL is `ADE XL Default`. You can also define a custom job policy, save it with a unique name in the `.cadence/jobpolicy` directory and set it as default by using the defaultJobPolicy environment variable.

The order in which the job policy files are searched in the `.cadence/jobpolicy` directory is determined by the Cadence Setup Search File mechanism (CSF). To find this information, CSF uses the `setup.loc` file, which is an ASCII file that specifies the locations to be searched and the order in which they should be searched. For more information about the setup.loc file or how to edit search order, see Cadence Setup Search File: setup.loc in the *Cadence Application Infrastructure User Guide*.

A job policy can be defined for:

- A single test. For more details, refer to Setting Up a Job Policy for a Test.
- A global job policy that is applied to all the tests that do not have their own job policies. For more details, refer to Setting Up the Default Job Policy on page 434.

While running a simulation, ADE XL looks for a job policy in the order given below and applies the policy that is found first:

- A test-specific job policy applied to the test by using the using the *Job Setup* context-sensitive menu command for that test.
- A global job policy applied to the ADE XL session by using the *Options — Job Setup* command.
- The default job policy specified by using the defaultJobPolicy environment variable in the `.csdenv` file.
- If no custom job policy setting is found, the `ADE XL Default` job Policy is used.

Important

If you want to use a custom job policy in every new adexl view, set the defaultJobPolicy environment variable to the name of your job policy. However, if you do not want to use the default job policy for a particular adexl view, you can apply a different policy to that view. The job policy information is saved in the active setup database of that view and is applied in every subsequent use.

Setting Up a Job Policy for a Test

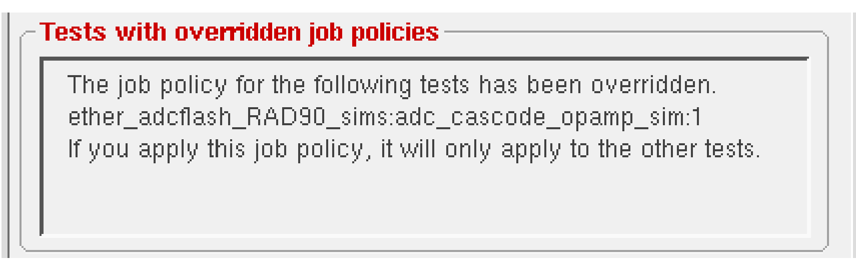
You can set up a job policy for each test. This enables you to use different distributed processing resources or queues for different tests.

➡ To set up a job policy for a test, on the Data View pane, right-click a test and choose *Job Setup*.

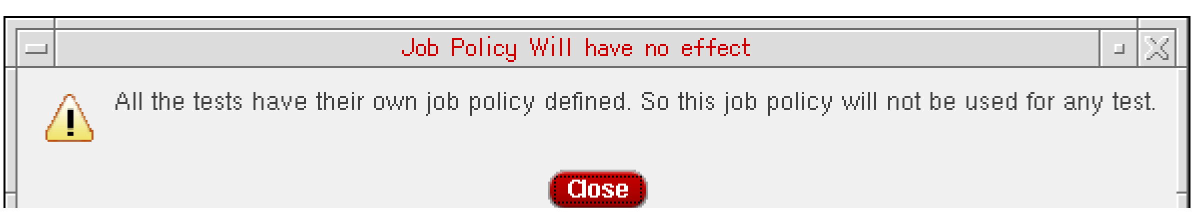
➡ To remove the job setup for a test and apply global job policy for it, on the Data View pane, right-click a test and choose *Clear Job Setup*.

Important Points Related to Job Policies

- Name of a test-specific job policy is shown in the test tooltip.
- The job policy specified for a test overrides the global job policy specified using *Options – Job Setup*. Information about the test(s) for which job policy is defined is shown in the global Job Policy Setup form, as shown below.



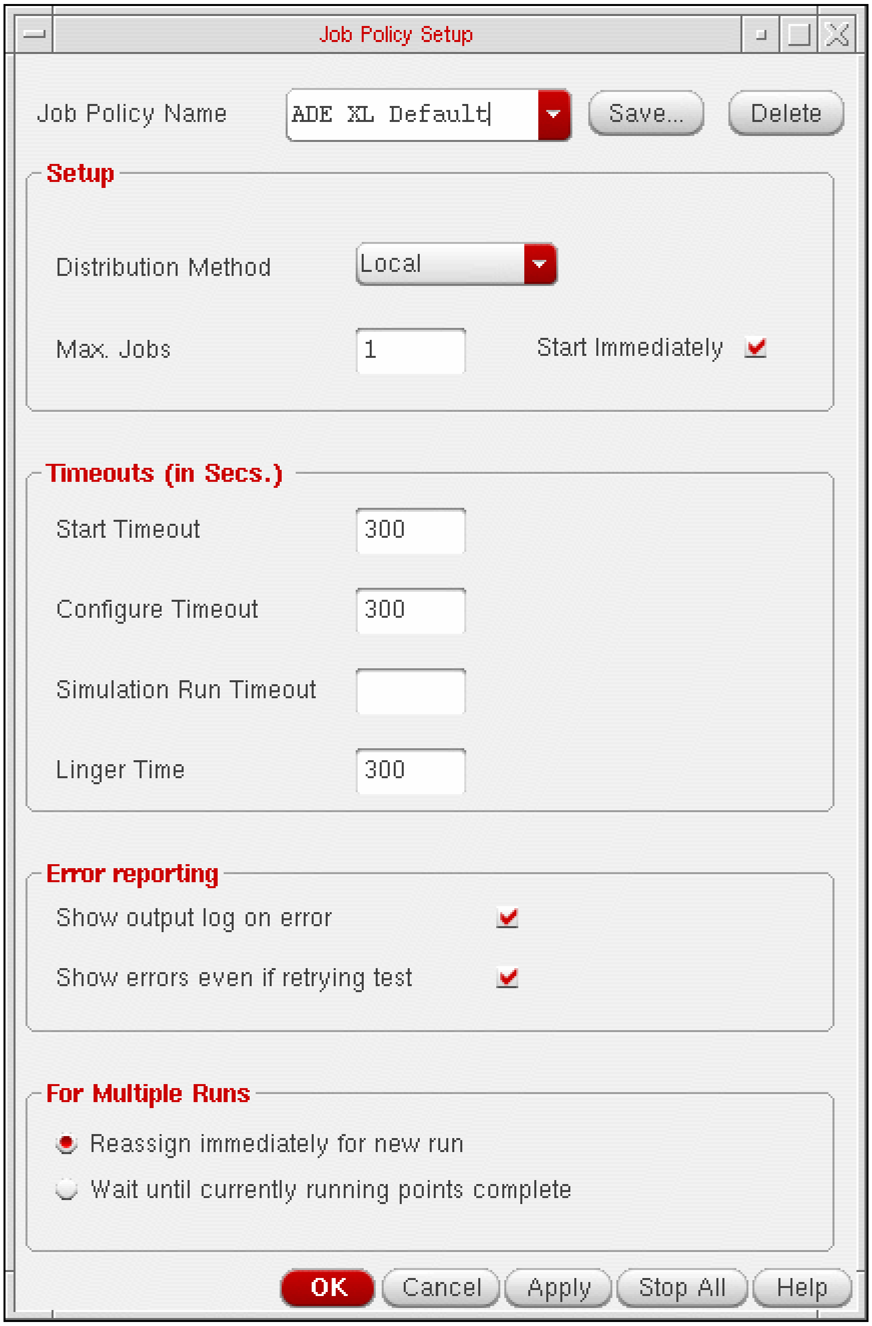
- If all the tests have test-specific job policy, the global job policy specified through the *Options – Job Setup* command will not have any impact. If you specify global job policy in this case, the following message is displayed.



- If some tests are running with test-specific job policies whereas other are running with global job policy, the *Options – Job Setup – Stop All* command stops the simulations for only those tests that are using the global job policy.
- For every test-specific job policy, a separate ICRP process is started and that is not used for any other test.
- If you change the job policy for a test while an ICRP process was running for that test, ADE XL completes the simulation for the running point. After that, the job is terminated and a new ICRP process is started according to the new job policy.

Setting Up the Default Job Policy

You cab specify a custom policy by using the Job Policy Setup form.



If you do not specify a job policy, the program applies the following defaults:

Distribution Method	Local
Max. Jobs	1
Start Timeout	300 (seconds)
Configure Timeout	300 (seconds)
Run Timeout	Infinite
Linger Time	300 (seconds)

See the following topics for more information:

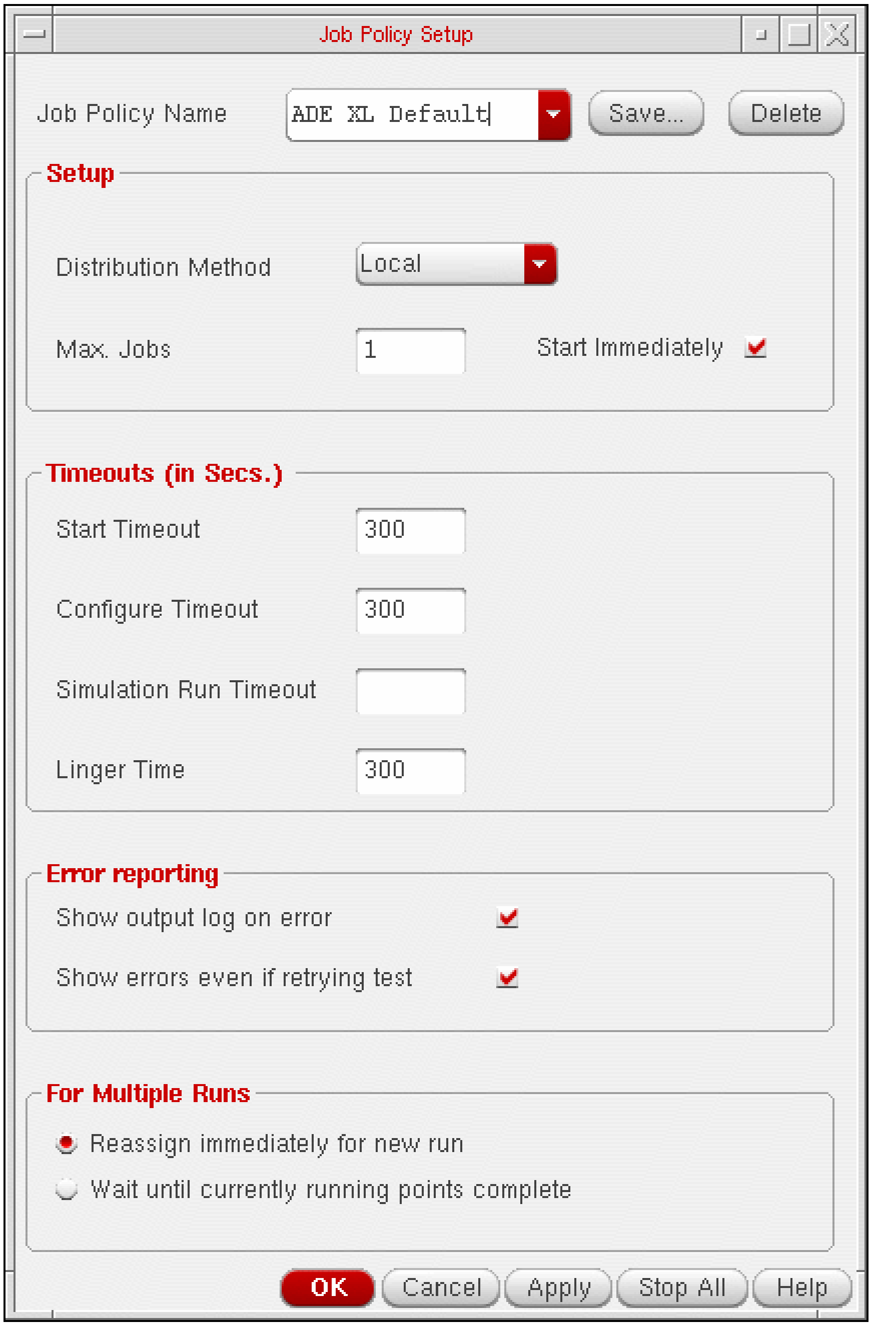
- Specifying a Job Policy Name on page 435
- Specifying a Distribution Method on page 437
- Specifying a Local Job Policy on page 437
- Specifying a Remote Host Job Policy on page 437
- Specifying a Command Job Policy on page 439
- Specifying an LBS Job Policy on page 441
- Specifying Max Jobs on page 444
- Specifying Job Timeouts on page 445
- Specifying Error Reporting Options on page 446
- Specifying Multiple Run Options on page 447
- Specifying a Job Submit Command on page 447
- Using Buttons at the Bottom of the Form on page 448
- Saving a Job Policy on page 449
- Deleting a Job Policy on page 450

Specifying a Job Policy Name

To specify a job policy name, do the following:

1. Choose *Options – Job Setup*.

The Job Policy Setup form appears.



2. In the *Job Policy Name* field, type a name for your job policy.

   The job policy name must be alphanumeric with no spaces or special characters.

   Note: You can specify the default job policy name by setting the defaultJobPolicy environment variable.

Specifying a Distribution Method

You can define your job policy using one of the following distribution methods:

- *Local* – The program runs simulations on the local host.
- *Remote-Host* – The program runs simulations on the remote host that you specify.
- *Command* – You send a command to the distributed processing software.
- *LBS* – The program uses Cadence Load Balancing Software.

Note: Both LBS and Command distribution method submit ADE XL job to a job distribution engine, for example Sun Grid Engine (SGE). In LBS mode, you can specify a selected set of qsub options whereas with the Command method you can submit a command with any qsub option.

Specifying a Local Job Policy

To specify a local job policy, do the following:

1. Specify a name for your job policy.
2. In the *Distribution Method* drop-down list, select *Local*.
3. (Optional) Specify maximum job information.
4. (Optional) Specify job timeouts.
5. (Optional) Specify error reporting options.
6. (Optional) Specify multiple run options.
7. Save your job policy.

   For more information, see Saving a Job Policy on page 449.
8. Click *OK*.

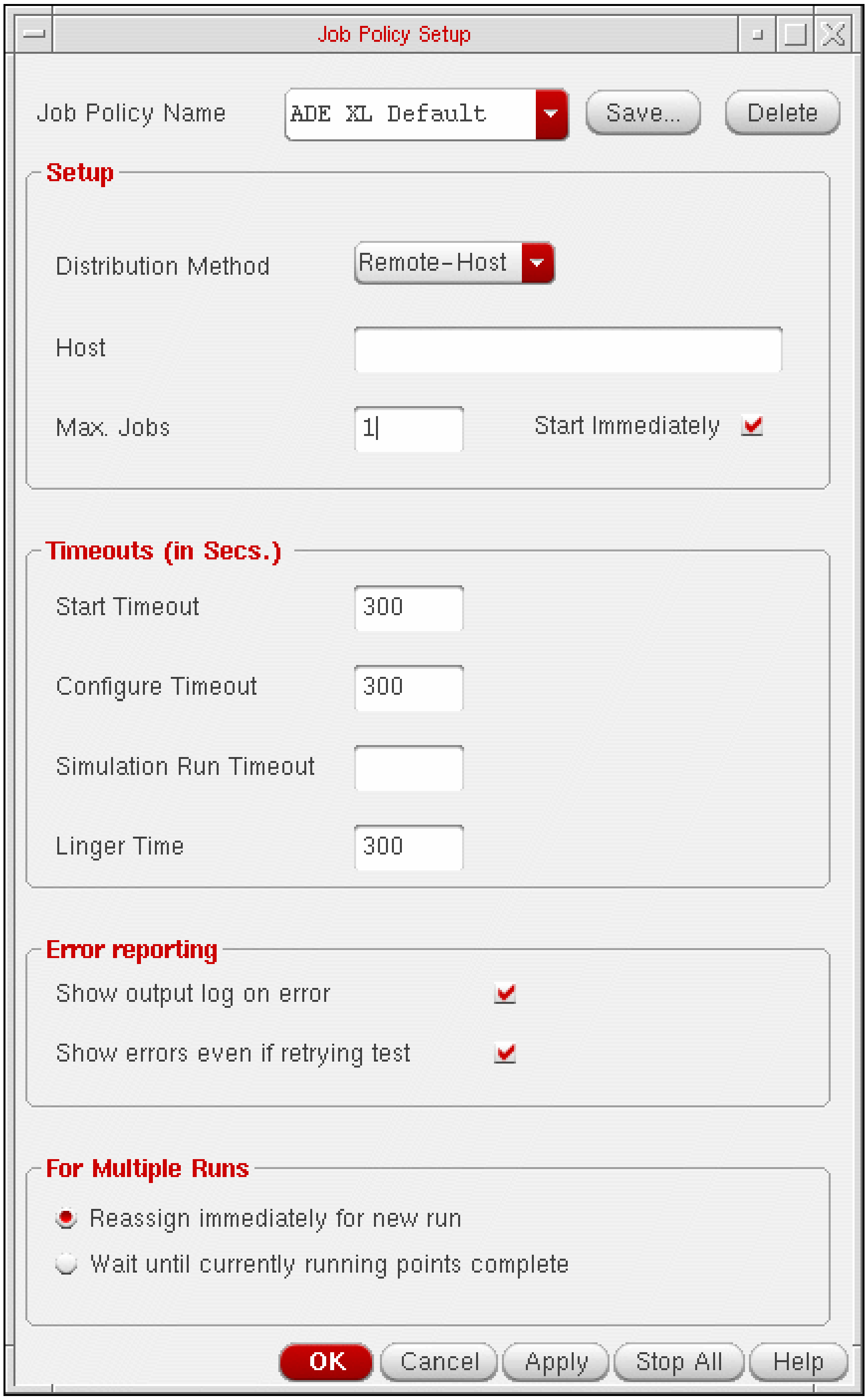
   The program runs simulations on the local host.

Specifying a Remote Host Job Policy

To specify a remote host job policy, do the following:

1. Specify a name for your job policy.
2. In the *Distribution Method* drop-down list, select *Remote-Host*.

   The *Host* field appears in the *Setup* group box.



3. In the *Host* field, type the name of a valid remote host.
4. (Optional) Specify maximum job information.
5. (Optional) Specify job timeouts.
6. (Optional) Specify error reporting options.
7. (Optional) Specify multiple run options.
8. Save your job policy.

   For more information, see Saving a Job Policy on page 449.
9. Click *OK*.

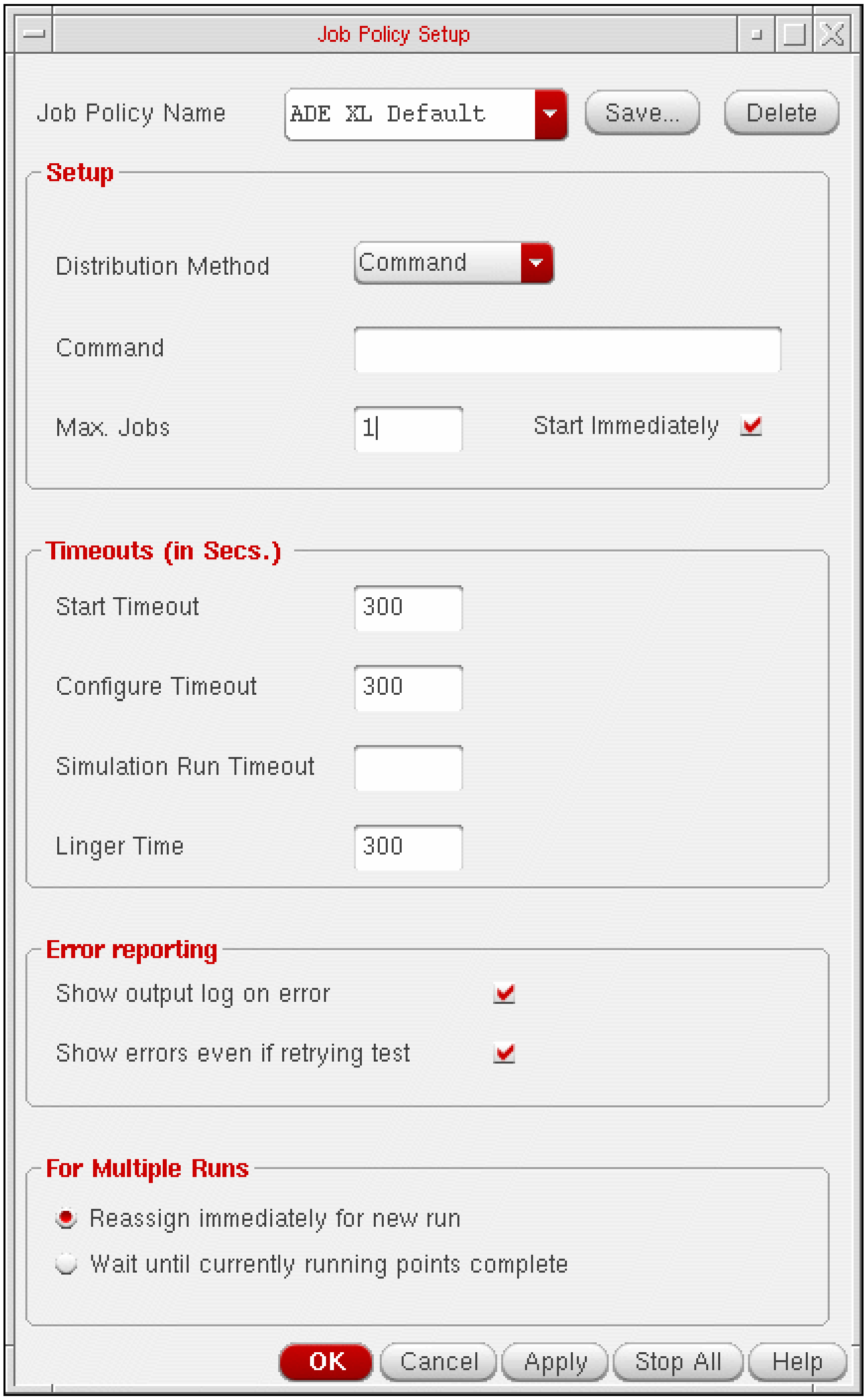
   The program submits remote simulations on the specified host.

Specifying a Command Job Policy

To specify a command job policy, do the following:

1. Specify a name for your job policy.
2. In the *Distribution Method* drop-down list, select *Command*.

The *Command* field appears in the *Setup* group box.



3. In the *Command* field, type the command you want to use to start jobs. The command must adhere to these guidelines.

4. (Optional) Specify maximum job information.
5. (Optional) Specify job timeouts.
6. (Optional) Specify error reporting options.
7. (Optional) Specify multiple run options.
8. Save your job policy.

   For more information, see Saving a Job Policy on page 449.
9. Click *OK*.

   The program uses the command to submit jobs.

Specifying an LBS Job Policy

ADE XL supports the following Distributed Resource Management Systems (DRMS):

LBS	Load Balancing System – Simple job distribution system (`cdsqmgr`) for setting up queues (collections of host machines) and hosts per queue; each queue has a job-per-host limit
LSF	Load Sharing Facility – DRMS from Platform Computing
SGE	Sun Grid Engine – Freeware from Sun Microsystems, Inc.

Note: See "Setting Up to Use Distributed Processing Option" in the *Virtuoso Analog Distributed Processing Option User Guide* for more information on setting up this job distribution method.

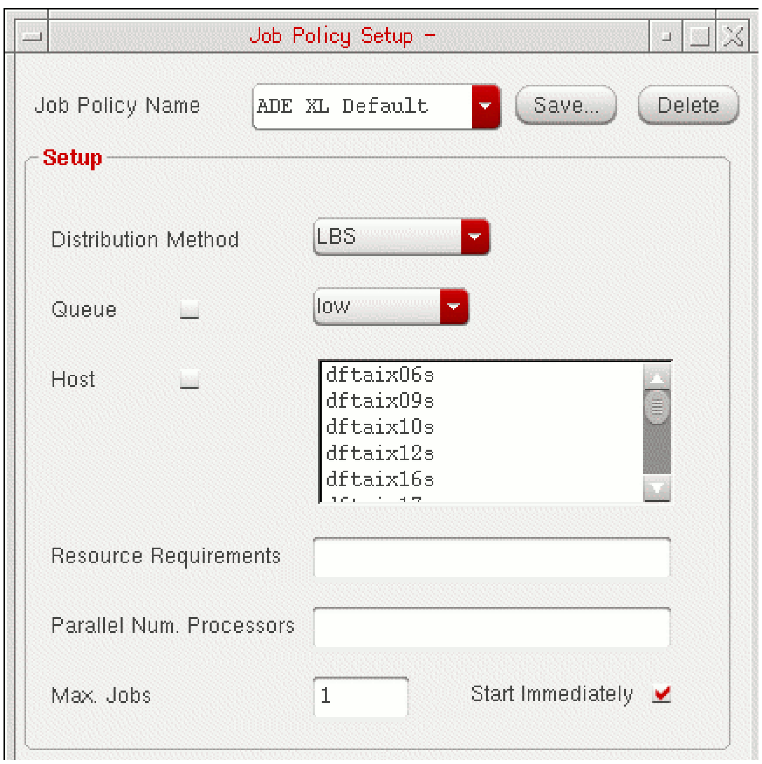
To specify a load balancing system job policy, do the following:

1. Specify a name for your job policy.
2. In the *Distribution Method* drop-down list, select *LBS*.

   Check boxes, fields, and list boxes related to specifying queues and hosts appear.

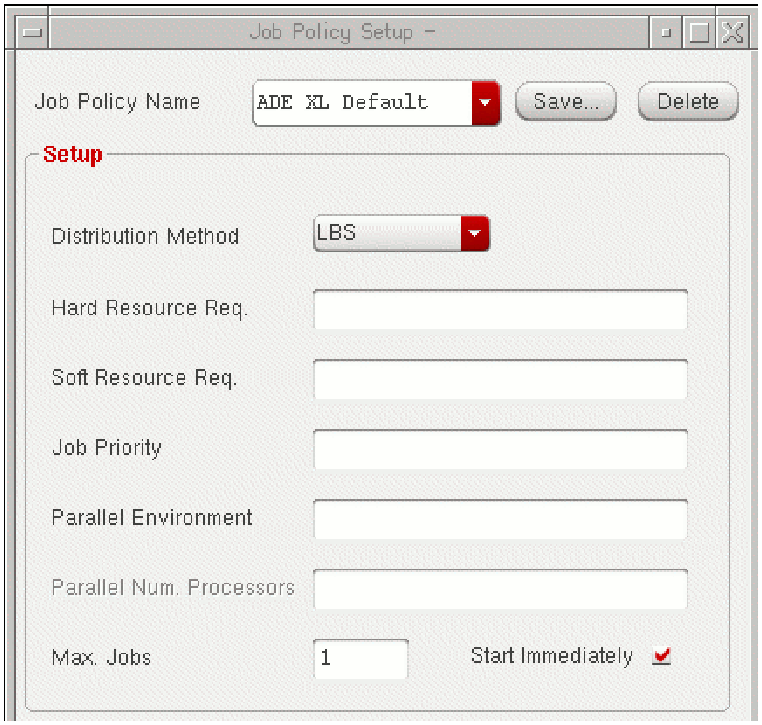
   If you have set up the LSF DRMS, the form shows the fields as shown below:

   Figure 13-1 Job Policy Setup Form for LSF



   If you have set up the SGE DRMS, the form shows the fields as shown below:

Figure 13-2 Job Policy Setup Form for SGE



3. (For LSF only) Select the *Queue* check box and select an available queue from the drop-down list.

   Note: Your system administrator determines the list of available queues. See "System Administrator Information" in the *Virtuoso Analog Distributed Processing Option User Guide* for more information.

   If you do not select the *Queue* check box, the program uses default system queue.

4. (For LSF only) Select the *Host* check box and select an available host from the list area.

   Note: The list of hosts corresponds to those available in the selected queue (see the previous step).

   If you do not select the *Host* check box, your DRMS decides which host to use.

5. (For LSF only) In the *Resource Requirements* field, specify any additional resource requirements to submit the job.

   The program uses these resource requirements along with the queue/host you specified to start the job. It is up to the load sharing facility software to resolve any conflicts between the queue/host and the resource requirements.

6. (For SGE only) In the *Hard Resource Req* field, specify the resources that must be allocated before a job can be started.

   For example, `num_proc=4,mem_total=4G`

7. (For SGE only) In the *Soft Resource Req* field, specify the resources that a job needs but do not have to be allocated before the job can be started. The specified resources will be allocated to the job on an as-available basis.

   For example, `mem_free=2G`

8. (For SGE only) In the *Job Priority* field, specify the priority for the job being submitted relative to other pending jobs submitted by you. Default priority of a submitted job is `0`. Users with administrator privileges can set this value from `-1023` to `1024`. Other users can set this value from `-1023` to `0`.

   For example, you can set priority for a job as `-500`.

9. (For SGE only) In the *Parallel Environment* field, specify the name of the parallel environment.

10. In the *Parallel Num Processors* field, specify the number of parallel processors to be used to run the submitted job.

11. (Optional) Specify maximum job information.

12. (Optional) Specify job timeouts.

13. (Optional) Specify error reporting options.

14. (Optional) Specify multiple run options.

15. Save your job policy.

    For more information, see Saving a Job Policy on page 449.

16. Click *OK*.

Specifying Max Jobs

Max jobs specifies the maximum number of ICRP jobs that can run in a ADE XL session. The Max Jobs value along with the value specified in the Specify field on the Run Options form is used by ADE XL to assign jobs to the simulations running in a session. If multiple

To specify maximum job information, do the following:

➤ In the *Max. Jobs* field, type the maximum number of jobs that can run at any time during your ADE XL session.

Note: If the distribution method specified in your job policy is *Command*, *Local* or *Remote-Host*, the maximum number of jobs that can run at any time cannot exceed the value specified using the maxIPCJobsLimit environment variable. The default value of this environment variable is `100`.

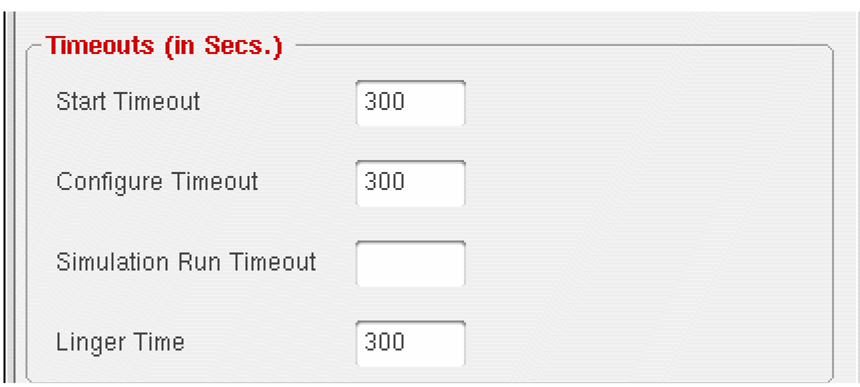
When *Start Immediately* is turned on, the program immediately submits the specified maximum number of jobs, or one job for every test, whichever is less, when you start ADE XL. This means that the defined number of ICRP processes are already running before you start the simulation. This saves the overhead of staring ICRP processes at time of running simulation.

To turn off this feature, do the following:

➤ Deselect the *Start Immediately* check box.

However, it is also important to understand that even if already started, the life of an idle ICRP process is controlled by the *Linger Time* field whose default value is `300` second. This implies that if a user launches a simulation 5 minutes after starting ADE XL, these processes would have already been killed and would restart when a simulation is run.

Specifying Job Timeouts



To specify job timeouts, do the following in the *Timeouts* group box:

1. In the *Start Timeout* field, type an integer number of seconds of time to wait for the `icrp` process to report back to ADE XL that it has started the job. The wait time starts as soon as ADE XL submits the job.

   If `icrp` does not respond in the number of seconds you specify, ADE XL considers the job not started and kills it. If there is another job waiting to start, ADE XL starts it.

2. In the *Configure Timeout* field, type an integer number of seconds to wait for the `icrp` process to report back to ADE XL that it has configured the job. The wait time starts as soon as ADE XL sends the job configure command.

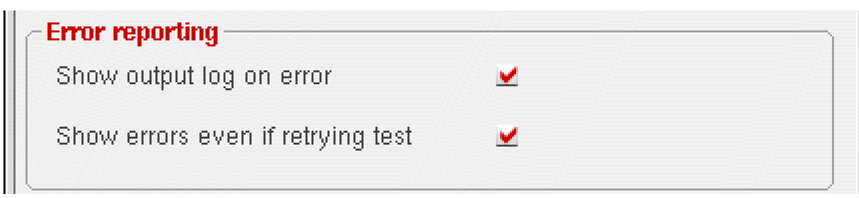
   If `icrp` does not respond with a successful configuration message in the number of seconds you specify, ADE XL considers the job configuration unsuccessful and kills it. ADE XL looks for the next idle job and tries to start and configure it.

3. In the *Run Timeout* field, type an integer number of seconds to wait for the `icrp` process to report back to ADE XL that it has run the job. The wait time starts as soon as ADE XL sends the run command for the job.

   If `icrp` does not respond that the job has completed in the number of seconds you specify, ADE XL considers the job run unsuccessful and kills it. ADE XL looks for the next idle job and tries to start, configure, and run it.

4. In the *Linger Time* field, type an integer number of seconds after which you want the program to kill a remote job (a remote workbench process) after the simulations finish.

Specifying Error Reporting Options



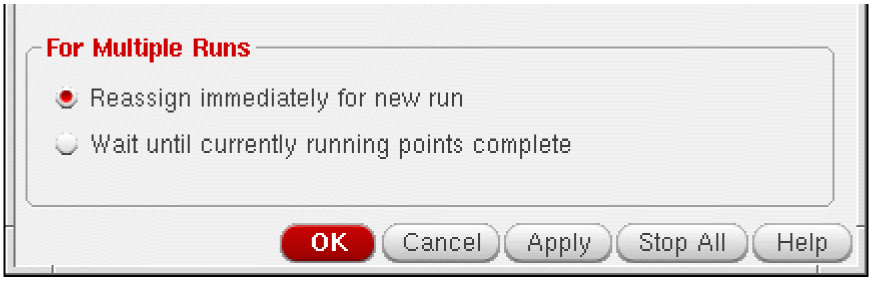
To specify error reporting options, do the following in the *Error Reporting* group box:

➤ By default, if multiple error points exist for a test, the program displays only the output log file of the first error point in the test. To view the output log file of the remaining error points, right-click the job status icon on the Run Summary assistant pane.

  To cause the program to display the output log files of all error points in the test, select the *Show output log on error* check box.

➤ To cause the program to display the output log file on the occurrence of an error for a test, even if the ADE XL distribution system is retrying the test, select the *Show errors even if retrying test* check box.

Specifying Multiple Run Options



To specify the job option to be used when running multiple runs in the same ADE XL session, do one of the following in the *For Multiple Runs* group box.

➤ To cause the program to reassign a completed job for the current run to a new run, select the *Reassign immediately for new run* option.

➤ To cause the program to wait until all the jobs for the current run are completed before assigning the jobs to a new run, select the *Wait until currently running points complete* option.

Also see: Setting Up Run Options

Specifying a Job Submit Command

When the software submits a job, it runs a separate process (`icrp`). The software prepends the command you type in the *Command* field on the Job Policy Setup form to the `icrp` startup command. The program runs the entire command on the local host.

Important

The command you type must take the `icrp` startup command as its argument.

For example, you might type a Distributed Resource Management System (DRMS) command such as `bsub` for LSF or `qsub` for Sun Grid Engine (SGE) in the *Command* field:

```
bsub -I -q queueName other_bsub_options
```

The program puts the `icrp` startup command at the end of the command you type as follows:

```
bsub -I -q queueName other_bsub_options icrpStartupCommand
```

You are responsible for making sure that this command is valid, that the `bsub` command you type can start `icrp` on some host in the cluster using the specified options.

Tip

You can try customizing your job submissions by developing wrapper scripts and using them with a command job policy to submit your jobs. The program will not validate the correctness of your wrappers/scripts.

Using Buttons at the Bottom of the Form

Use the buttons at the bottom of the form as follows:

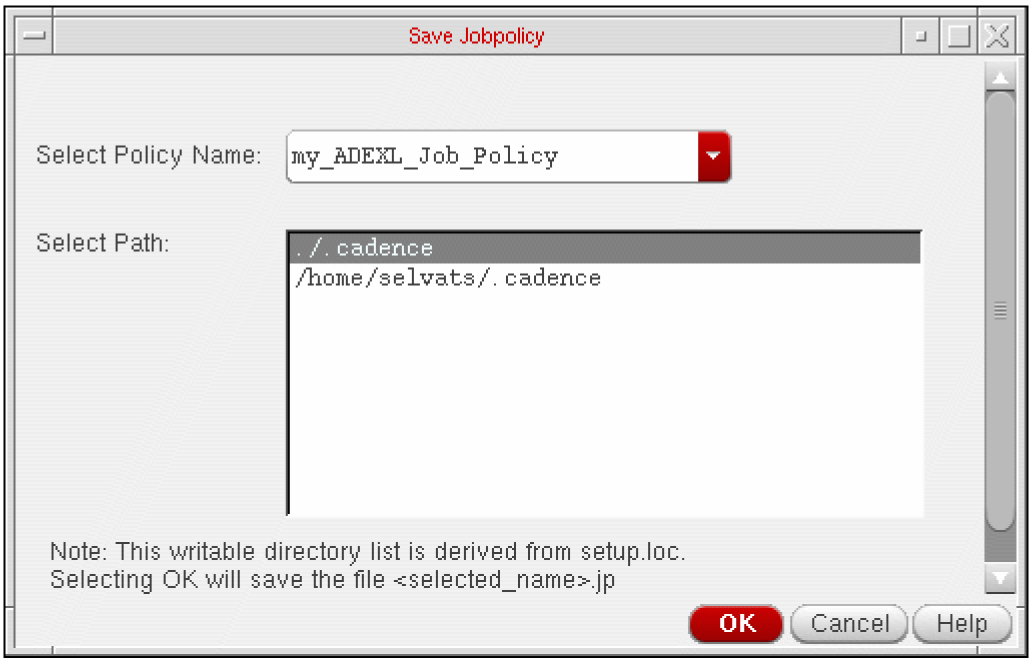
Button	Description
OK	Saves the settings for the job policy, applies it to all future jobs you start during the current session, and closes the form **Note:** The program kills any currently running jobs as soon as they become idle.
Cancel	Closes the form without saving your changes to the settings for the job policy
Apply	Saves the settings for the job policy and applies it to all future jobs you start during the current session **Note:** The program kills any currently running jobs as soon as they become idle.
Stop All	Stops all the jobs you started during the current session regardless of their current state (started, preparing UI, populating results etc.) *Important* Use this option with extreme caution.
Help	Opens the online Help page for this form

Saving a Job Policy

To save a job policy, do the following:

1. In the Job Policy Setup form, click *Save*.

   The Save Job Policy form appears.



2. In the *Select Path* list, select the directory where you want to save the job policy.

   You can save the job policy in one of the following directories in which you have write permissions or in the paths specified in your `setup.loc` file where you have write permissions:

   - `.cadence` folder in the current directory
   - The `.cadence` folder in the path specified in the `CDS_WORKAREA` environment variable.
   - `$HOME/.cadence` (the `.cadence` folder in your home directory)
   - The `.cadence` folder in the path specified in the `CDS_PROJECT` environment variable.
   - The `.cadence` folder in the path specified in the `CDS_SITE` environment variable.

3. Click *OK*.

The job policy is saved in the `jobpolicy` folder under the selected directory. The job policy file has the `.jp` extension.

If a job policy file with the same name is found in more than one of the above locations, the first job policy file found in the first of the following locations is used:

- `.cadence` folder in the current directory
- The `.cadence` folder in the path specified in the `CDS_WORKAREA` environment variable.
- `$HOME/.cadence` (the `.cadence` folder in your home directory)
- The `.cadence` folder in the path specified in the `CDS_PROJECT` environment variable.
- The `.cadence` folder in the path specified in the `CDS_SITE` environment variable.

Deleting a Job Policy

To delete a job policy, do the following:

1. Choose *Options – Job Setup*.

   The Job Policy Setup form appears.

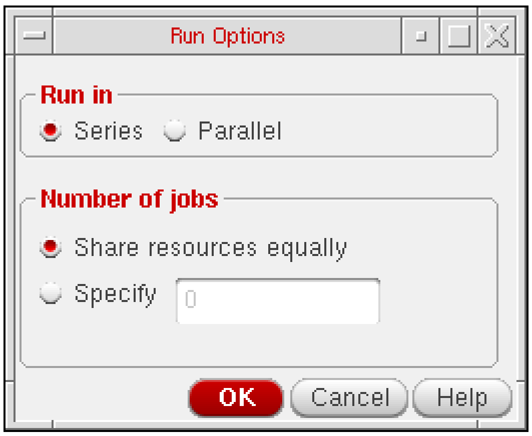
2. From the *Job Policy Name* drop-down list, select the job policy you want to delete.
3. Click *Delete*.

Setting Up Run Options

When you perform multiple ADE XL runs simultaneously (that is, an ADE XL run is already in progress and you click  again to start more runs simultaneously), the options specified on the Run Options form are used to allocate the available ICRPs or jobs. You can change the options on this form to control allocation of the ICRPs among various ADE XL runs.

To set up run options, in the ADE XL window, choose *Options – Run Options*.

The Run Options form appears, as shown below.



The following table describes the commands available on the Run Options form:

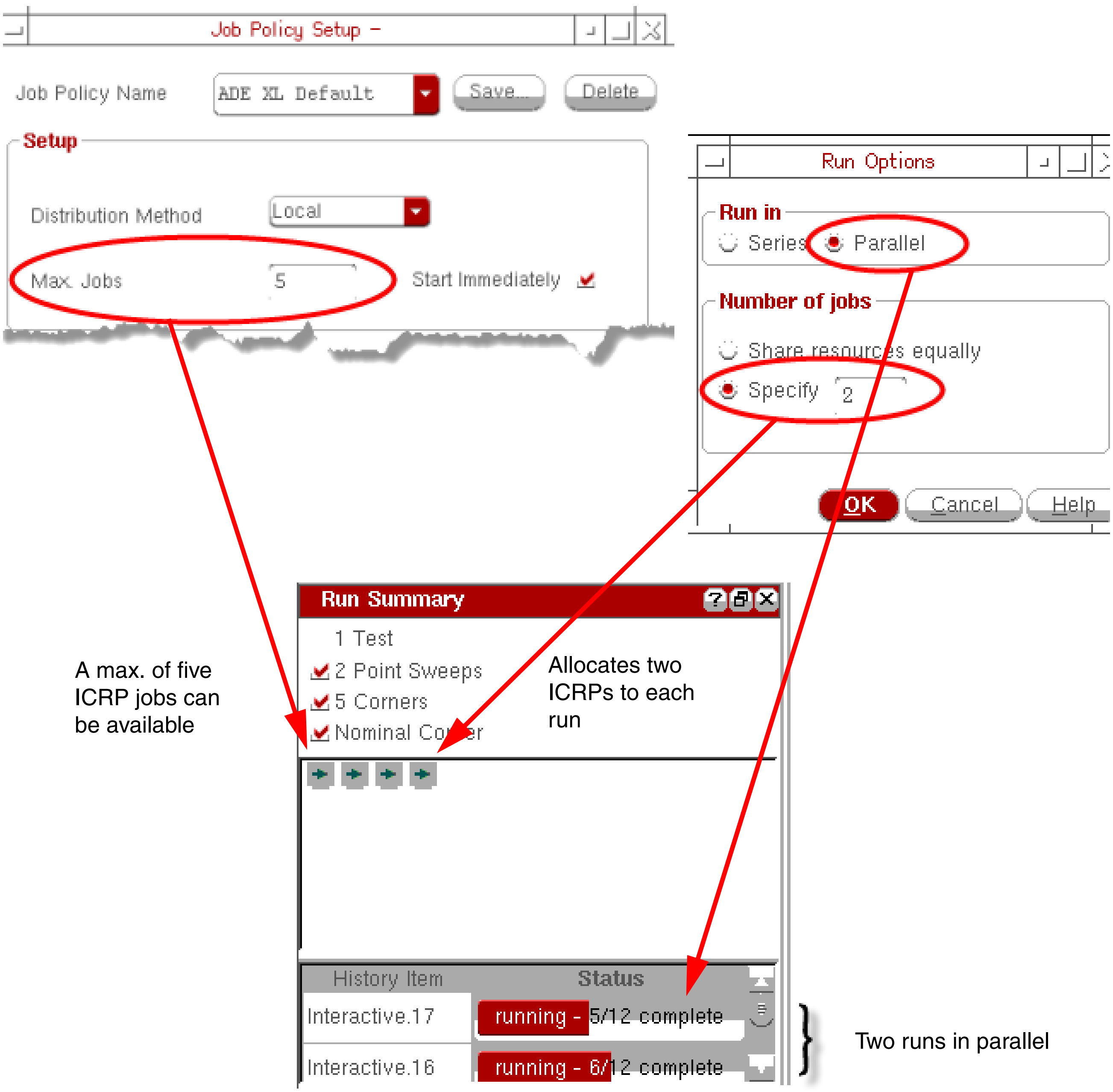
Command	Description
Run In group box	Contains the options to specify how more than one ADE XL runs are to be performed
Series	Performs multiple ADE XL runs in series. In this case, only one run is performed at a particular point of time. If the number of available ICRPs (as specified by *Max Jobs* in the job policy) is greater than the number of points in the current ADE XL run, only the ICRPs required to run the points are used. The remaining ICRPs are not allocated to any subsequent run until the current ADE XL run is complete.

Command	**Description**
Parallel	Runs multiple ADE XL runs in parallel. If ICRPs are available, multiple ADE XL runs are started without waiting for the previous one to complete. The allocation of points or corners to ICRPs depends on the setting in the *Number of jobs* section on this form. **Note:** The default value for *Run In* is set to *Series*. To make *Parallel* as the default value, set the defaultRunInParallel environment variable to `t`.
Number of jobs group box	Contains the options to specify how the ICRPs are to be allocated among multiple ADE XL runs.
Share resources equally	When *Run In* is set to *Parallel*, this option shares the number of jobs among all the parallel runs equally. For example, if *Max Jobs* is `10` and the current run is using `4` ICRPs, two additional parallel runs will each get `3` ICRPs.
Specify	When *Run In* is set to *Parallel*, this option specifies the maximum number of ICRPs to be allocated to each ADE XL run in parallel. **Note:** The value in the *Specify* field should not exceed the value of *Max Jobs* because in that case, the value in *Specify* is ignored. If the value of *Max Jobs* is greater than the value in *Specify*, each ADE XL run gets the same number of ICRPs as specified in the *Specify* text box. **Note:** You can change the value of *Specify* between two ADE XL runs. For example, *Max Jobs* is set to `10`. Before starting the first run, *Specify* is set to 4, so it gets four ICRPs. After this, if you change the value of *Specify* to 2 and start two more runs, the subsequent runs will get two ICRPs each. Also see: Examples of ICRP Allocation to Parallel Runs Based on Specify and Max Jobs

Examples of ICRP Allocation to Parallel Runs Based on Specify and Max Jobs

Example 1

If *Max Jobs*=5 and *Specify*=2, the first ADE XL run will get two ICRPs and the subsequent runs will also get two ICRPs allocated to them. If there are only two runs in parallel, one ICRP still remains unallocated, as shown in the figure given below.



Example 2

If *Max Jobs*=`6` and *Specify*=`4`, the first ADE XL run will get four ICRPs. The next run initially gets the remaining two ICRPs. Depending on the value set for *Reassign immediately for new runs* job policy setting, ADE XL reassigns one of the ICRPs from the first run to the second run so that each run is using three ICRPs.

Example 3

If *Max Jobs*=`24` and *Specify*=`8`, the first run gets eight ICRPs. The next run will get another eight ICRPs. Eight more ICRPs still remain unallocated.

Running Pre-run Scripts before Simulation Runs

For each test, you can specify a pre-run script that contains a set of OCEAN commands to be run before the test is simulated. If a pre-run script is specified for an ADE XL test, for every point of evaluation, first the pre-run script is run and then the test is simulated for that point.

Using pre-run scripts, you can specify OCEAN commands that read the simulation setup for a test, run pre-simulation analyses for each simulation point and, if required, modify the simulation setup for that point. For example, you can run a script to calculate a calibrated value for a design variable and use that value in the main simulation run. To see an example, refer to Pre-run script Example for Monte Carlo Calibration.

In a pre-run script, you can use any OCEAN command. For example, you can use the `desVar` command to get the value of a design variable in an ADE XL test. To see how, refer to Pre-run Script Example. In addition, you use the following OCEAN commands that are specifically meant for use in pre-run scripts:

- ocnxlLoadCurrentEnvironment
- ocnxlSetCalibration
- ocnxlRunCalibration
- ocnxlAddOrUpdateOutput
- ocnxlUpdatePointVariable
- ocnxlGetJobId
- ocnxlGetPointId
- ocnxlMCIterNum
- ocnxlMainSimSession

For more information about these and other OCEAN commands, see the *OCEAN Reference*.

Note: A pre-run script is run in non-graphical mode. Therefore, in this script, you cannot specify OCEAN commands that plot waveforms.

For more information, see the following topics:

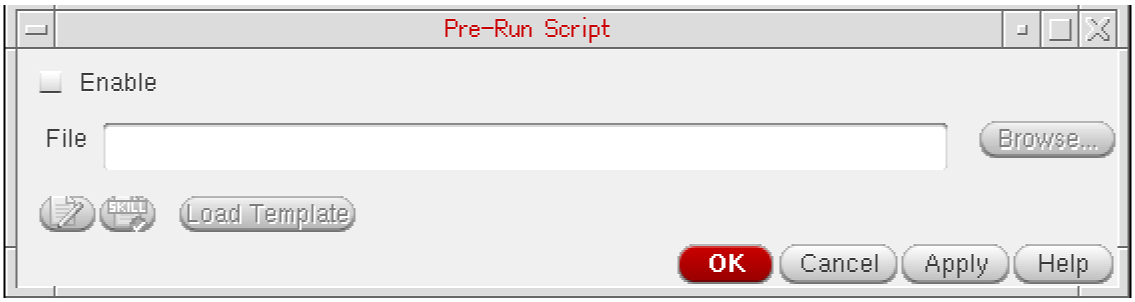
- Adding a Pre-Run Script on page 456
- Modifying a Pre-Run Script on page 459

Adding a Pre-Run Script

To add a pre-run script for a test, do the following:

1. In the Data View pane, right-click the test and choose *Pre-Run Script*.

   The Pre-Run Script form appears.

2. Click *Enable* to enable the use of a pre-run script for the test.

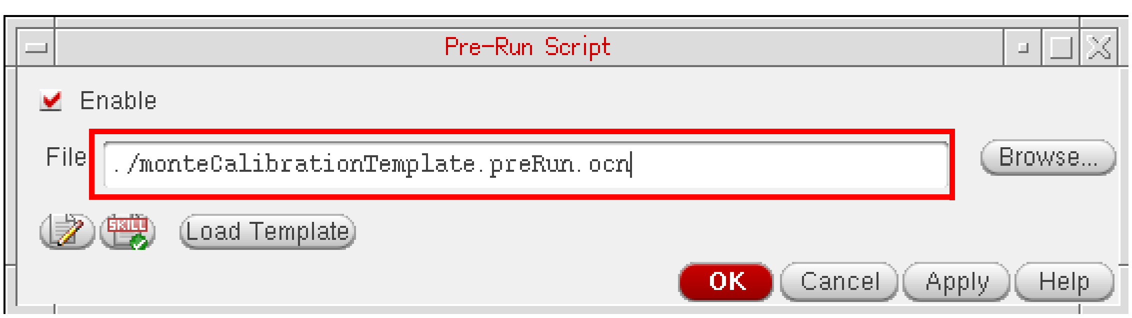
   All other fields and buttons on the form are enabled.

3. Do any of the following to provide a pre-run script:

   ❑ If a script is already saved in a file, specify the path to the file in the *File* field. You can click *Browse* to view the directories and select a file.

   Now, click (*Edit pre-run script*) to open and edit the script in the default text editor for Virtuoso specified by using the `$VISUAL` or `$EDITOR` variables in the `.cshrc` file, in the given order of preference. If none of these two variables is set, the tool uses the vi editor.

   ❑ To load a script template file and use it to create a new script, click Load Template. The path to the default template, `monteCalibrationTemplate.preRun.ocn`, is displayed in the *File* field.

   You can now click *Edit pre-run script* to open and edit the script in the default text editor.

4. After editing, save the script and close it.

   If you used a script template, save your script with a new name.

5. (Recommended) Click (*Lint check pre-run script*) to run SKILL Lint rules and check the scripts for errors.

   The script is displayed in the SKILL IDE where you can run lint rules and modify your script.

6. Save the script and close the SKILL IDE.

7. Click *OK* to set the script for your test.

You can also run the following SKILL fucntions to set a pre-run script:

- axlSetPreRunScriptEnabled
- axlImportPreRunScript
- axlSetPreRunScript
- axlGetPreRunScript

Note: By default, the pre-run script file specified for a test is copied to the ADE XL view and is saved as a part of the state. To use the script file from its original location and not to copy it to the ADE XL view, set the copyPreRunScripts environment variable to `nil`.

The following examples show how pre-run scripts can be used.

Example 13-1 Pre-run Script Example

The following pre-run script reads the simulation setup of a test and before every point of evaluation, it runs a transient simulation for a selected set of sweep values of design variable `R1`. Based on the simulation result, it updates the value of another variable `DC1` before running the main simulation for that point.

```
; Read the simulation setup for the test
ocnxlLoadCurrentEnvironment(?noAnalysis t)

; If design variable R1 is swept from 10K to 30K, the following logic will run only
; for the points where R1 > 20K
when(evalstring(desVar(“R1”)) > 20K
    resultsDir(“./newDir”)
    save( 'i "/NM2/D" "/NM0/D" )
        analysis( ‘tran ?stop “10u”)
        run( )
    SimRes=xmax(i("NM2.d" ?result "tran-tran" ?resultsDir "./newDir/psf") 1 ))
```

```
; Update value of variable DC1 in the simulation setup for the point based on
; whether R is greater than or lesser than 90u.
    a=110u
    b=120u
    if( SimRes > 90u then
    ocnxlUpdatePointVariable("DCI" sprintf( nil "%L" a))
    else
    ocnxlUpdatePointVariable("DCI" sprintf( nil "%L" b))
    )
)
```

Example 13-2 Pre-run script Example for Monte Carlo Calibration

The following pre-run script customizes the simulation setup of a test and sets up a single iteration Monte Carlo calibration run by inheriting the settings from the main Monte Carlo Sampling run. Next, it calculates a calibrated value that is used in the main simulation.

In the Pre-Run Script form, click *Enable*. Next, click *Open Template*. ADE XL loads the *monteCalibrationTemplate.preRun.ocn* template.

Click (*Edit pre-run script*) to open and edit the script in the default text editor for Virtuoso. The following template script is opened.

```
; Read the simulation setup for the test Test1
ocnxlLoadCurrentEnvironment( ?noAnalysis t)


; The following command changes the simulation setup for the calibration run by
; specifying a new analysis for the calibration run
    analysis('ac ?start "1G"  ?stop "3G"  ?step "1M")


; The following command sets up a single iteration Monte Carlo calibration run by
; inheriting the settings from the main Monte Carlo Sampling run
    ocnxlSetCalibration()


; The following logic calculates the calibrated value using the successive
; approximation method. The ocnxlRunCalibration() runs a single iteration Monte
; Carlo simulation.
    for( n 1 noOfBits
        i = noOfBits - n
        BitWord = BitWord + 2**i
        ParamVal = BitWord*8
        desVar( "ParamName" ParamVal )
        ocnxlRunCalibration()
        SimResult = <Evaluate Expression>
        if(  SimResult < target
            then
            BitWord = BitWord - 2**i
```

```
        )
    )
    CalResult = BitWord * 8

; The following command adds the calibrated value as an output named
; Calibrated_ParamName so that it can be viewed in the ADE XL Outputs Setup tab for
; each point.
    ocnxlAddOrUpdateOutput("Calibrated_ParamName" CalResult)

; The following command updates the value of global variable ParamName with
; the calibrated result.
    ocnxlUpdatePointVariable("ParamName" sprintf( nil "%L" CalResult))
```

After assigning the calibrated values to the output of a test, `Test1` in the above example, you can also pass the calibrated value to a global variable in another test, for example, to variable `N2` in `Test2`. In this case, you can define variable `N2` for `Test2` using the `calcVal` function, as shown below:

```
calcVal("Calibrated_ParamName" "Test1")
```

Important

It is not possible to distribute a simulation running from a pre-run script. If you specify distributions setup in the script or in `.cdsenv`, it is ignored. The simulations in pre-run scripts are distributed with jobs as per the job policy set in ADE XL.

Modifying a Pre-Run Script

To modify the pre-run script for a test, do the following:

1. In the Data View pane, right-click the test and choose *Pre-Run Script*.

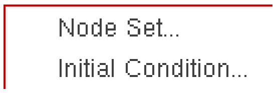
   The Pre-Run Script form appears.

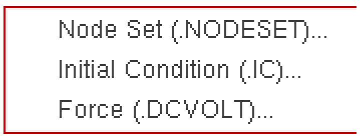
2. Verify that the *File* field shows the path to the required script file name.

3. Click  (*Edit pre-run script*).

   The script file is opened in the default text editor where you can edt the script and save the changes.

Using Convergence Aids





To specify node set information, do the following:

1. On the Data View assistant pane, right-click the test or analysis name and choose *Convergence Aids – Node Set*.

   The appropriate form for your target simulator appears.

2. When you are done specifying node set information, click *OK*.

To specify initial conditions, do the following:

1. On the Data View assistant pane, right-click the test or analysis name and choose *Convergence Aids – Initial Condition*.

   The appropriate form for your target simulator appears.

2. When you are done specifying initial conditions, click *OK*.

(hspiceD only) To force a node to a particular voltage for the entire simulation, do the following:

To specify node set information, do the following:

1. On the Data View assistant pane, right-click the test or analysis name and choose *Convergence Aids – Force*.

   The Select Force Node Set form appears.

2. When you are done specifying force nodes, click *OK*.

Disabling and Enabling All Point Sweeps

To disable all point sweeps, do the following:

➤ On the Run Summary pane, deselect the *# Point Sweeps* check box (where # indicates the number of point sweeps).

The simulator does not run any point sweeps.

To restore all point sweeps, do the following:

➤ On the Run Summary pane, select the *# Point Sweeps* check box (where # indicates the number of point sweeps).

The simulator runs the specified number of point sweeps.

Starting a Simulation

To start a simulation, do the following:

➤ On the Run toolbar, click the *Run Simulation* button.

The program does the following:

- ❑ Checks for new design variables by performing a copy-from-cellview operation.

  The simulator requires a value for every design variable or a global variable that is enabled and has a valid value or set of values.

- ❑ Generates a new netlist that reflects any design changes (any changes you made to the cellview since the last extraction).

  By default, ADE XL creates and saves a separate netlist file in the results directory for every design point. For large designs, this results in consuming huge space with same netlist file being saved in multiple directories. Starting this release, you can choose to save a single netlist for all the design points by setting the singleNetlistForAllPoints environment variable to t. When this variable is set, a single netlist file is created and a link to that is created in all the point directories.

- ❑ Updates the simulation information numbers that appear on the Run Summary assistant pane to indicate the number of tests running, the number of values varying (*Point Sweeps*), the number of corners, and the nominal corner.

- ❑ Displays the progress of the simulation run in the Run Summary assistant pane.

  For more details of the progress bar, refer to Checking Run Status on the Progress Bar.

- ❑ Simulates your testbench.

- ❑ Displays the progress and results of the simulation run on a new tab in the Results tab of the Outputs pane. The name of the tab matches the name of the checkpoint for the simulation run. For more information, see Working with Tabs for Simulation Checkpoints on page 561.
- ❑ If simulations are not run successfully, displays appropriate errors. For more information on the types of errors, refer to Simulation Errors on page 492.

After the simulation is complete, the results appear on the Results tab. For details on how to view results on the Results tab and how to customize the view, refer to Viewing, Plotting, and Printing Results.

In some cases, the results status is shown as `Canceled`. This status indicates any one of the following scenarios:

- You have canceled the simulation run by clicking the *Stop Simulation* button.
- The ADE XL GUI is closed abruptly.
- There is no sufficient disk space to complete the simulation run.
- If you are running a Monte Carlo simulation, a particular simulation is canceled by ADE XL because the target yield specified by selecting the *Auto Stop using Significance Test* field is met. For more details, refer to Stopping Monte Carlo Based on Target Yield.
- If you are running a Monte Carlo simulation, the *Run Nominal Simulation* option is enabled on the Monte Carlo setup form, but the Monte Carlo run for the nominal simulation fails.

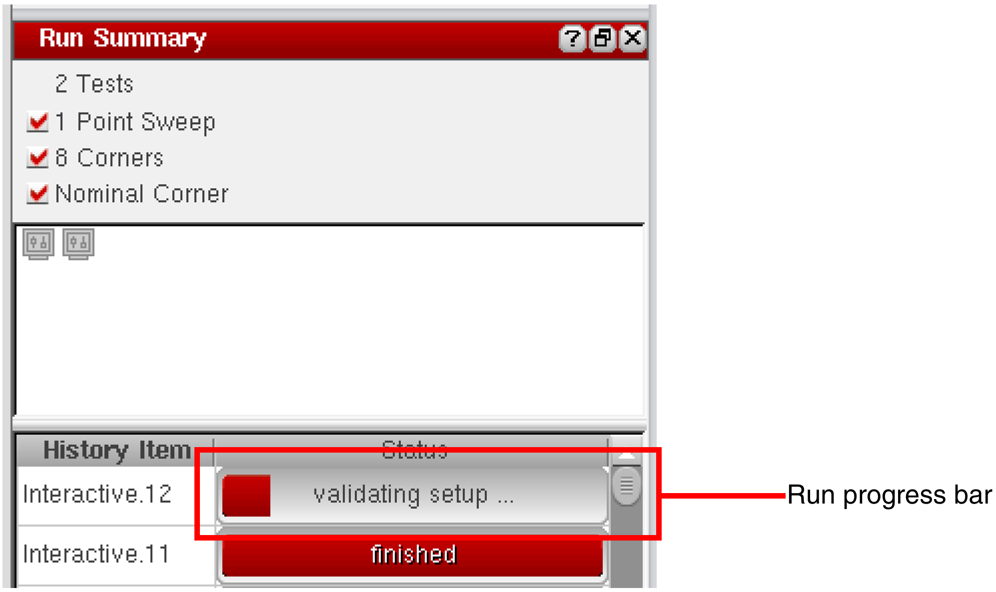
See also

- Checking Run Status on the Progress Bar on page 465
- Specifying the Run Mode on page 71
- Stopping a Simulation Run on page 466
- Stopping Jobs and Resubmitting Simulations on page 471
- Ignoring Design Changes During Run on page 473
- Continuing the In-Process Simulations After ADE XL GUI Exits on page 475
- Disabling and Enabling All Point Sweeps on page 461
- Disabling and Enabling the Nominal Corner on page 243
- Disabling and Enabling All Corners on page 245
- Simulating Corner on page 262

- Simulation Errors on page 492
- Creating and Running an OCEAN Script on page 535
- Chapter 14, "Viewing, Plotting, and Printing Results"

Checking Run Status on the Progress Bar

While ADE XL is preparing to run a simulation or during a run, you can check its status on the progress bar displayed on the Run Summary assistant pane. The progress bar is displayed to the right of the history item name, as shown in the following figure.



It shows one of the following status values to correctly indicate the step being performed:

Table 13-1 Status Values Displayed on the Run Progress bar

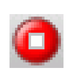
Status Value	Indicates that...
validating setup	Verification of the test setup, variables, parameters, and corners specified in the Data View pane is in progress. ADE XL is also verifying the run options specified for different types of runs in the corresponding run options form. If there are any errors or warnings, appropriate messages are displayed in the CIW or error message dialogs.
saving history	Creation of the test setup data is in progress. That is, ADE XL is creating the directories in which the test setup data for the simulation run will be saved.
preparing result directories	Creation of the results database (.rdb file) and the results directory is in progress.

Table 13-1 Status Values Displayed on the Run Progress bar

Status Value	Indicates that...
loading tests x/n	ADE XL is starting a session for each active test in the setup. Here `n` specifies the total number of active tests in the setup, not the total number of tests.
evaluating design points x/n	Creation and evaluation of the xth number of design point out of n design points is in progress. Here, a total of n design points is calculated based on the given variables, parameters, and run options. **Note:** These are not data points, but the design points that are created by sweeped values.
populating results UI	Preparation of the user interface to display results is in progress. After this step is complete, the *Results* tab for the current run is displayed on the ADE XL window. At this time, the run status for all the simulations is shown as *pending*.
running – x/n complete	ADE XL is running the `x`th run out of a total of `n` runs. Note that the progress bar is reset in this stage.
finished or *finished with errors*	Successful or unsuccessful completion of all the simulation runs.

Stopping a Simulation Run

To stop a simulation run, do one of the following:

- On the Run toolbar, click the *Stop Simulation*  button.
- On the Results tab of the Outputs pane, right-click on the name of the tab for the checkpoint for the simulation that is running and choose *Stop Run*.
- On the Run Summary pane, right-click on a history item and choose *Stop All* to stop all the jobs you started during the current session regardless of their current state (started, getting configured, running).
- Switch to the Status view on the Results tab and click the *Stop* button.

ADE XL stops all the simulations that are running. Results for the points that are already successfully complete are saved in the results database for the history checkpoint.

Important Points to Note

- Instead of stopping the entire simulation run, you can stop the simulation for only selected tests or points. For more details, refer to Canceling Simulations for Selected Tests or Corners.
- If required, you can rerun the simulation for incomplete points later. For more details, refer to Simulating Only Error or Incomplete Points.

Important

If you are using the Spectre circuit simulator, you can release your license by clicking the  button.

Canceling Simulations for Selected Tests or Corners

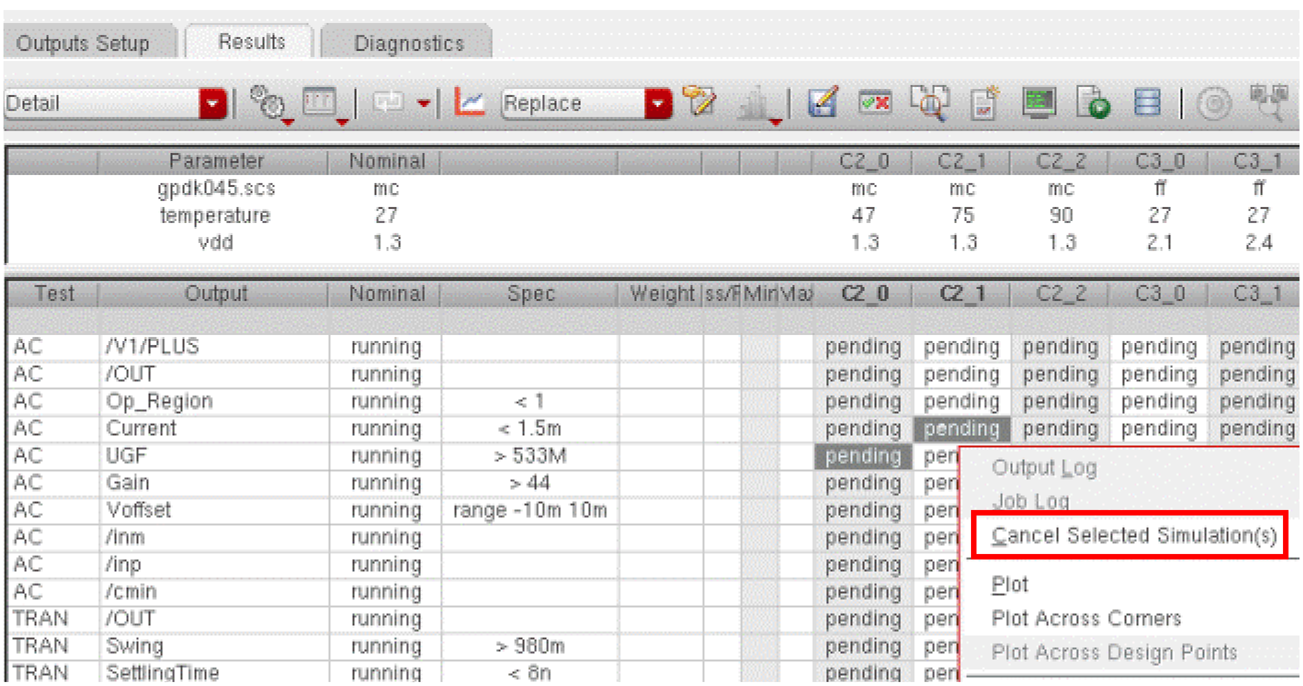
In certain cases, you might need to cancel the simulation for a specific test or corner and complete the simulation for the remaining. This is generally helpful in the following scenarios:

- When the results of a corner indicate that there is scope of improvement in the design and you do not want to run similar corners.
- When the results of a test indicate that the setup needs to be changed and you want to cancel all of the pending runs for that test, but complete the simulations for the other tests.
- When you are running resource-intensive simulations and you want to allow only the jobs that are already running to continue, and to cancel all the pending jobs so that the limited MMSIM licenses can be released for other high-priority simulations.

Canceling Simulations for Selected Corners

To cancel a simulation for one or more corners, do the following:

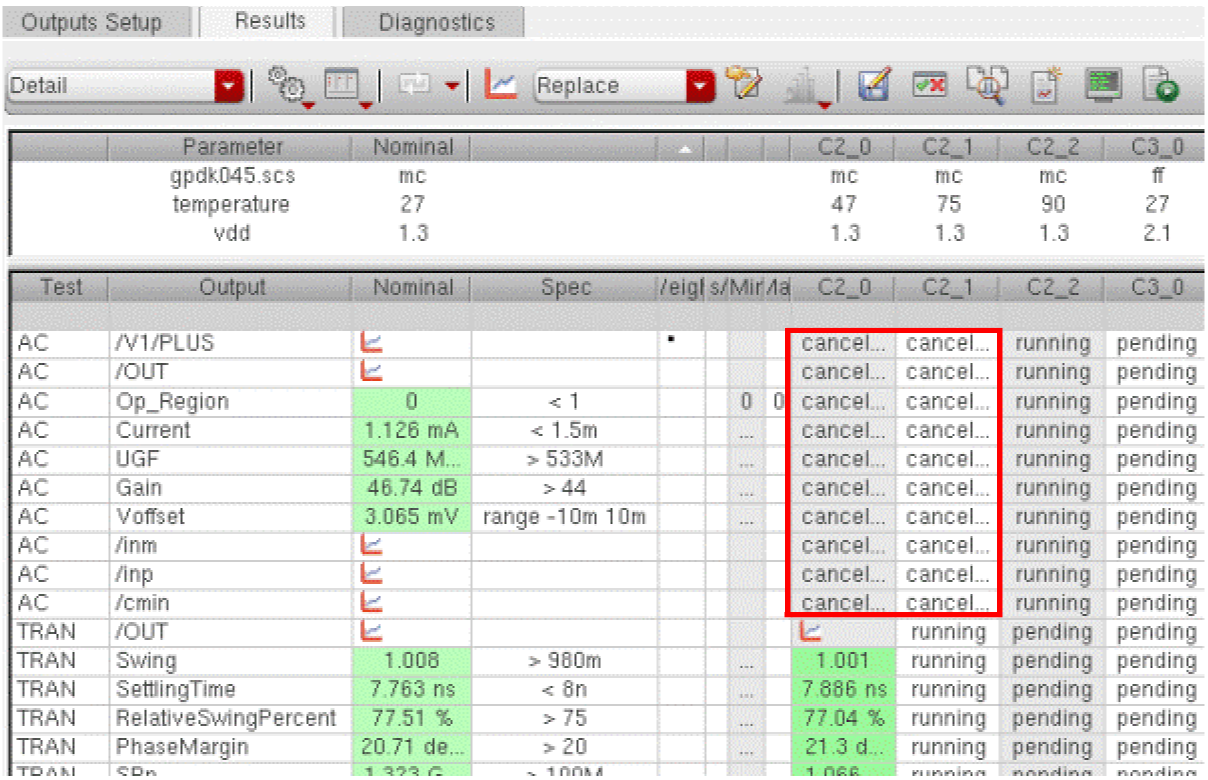
1. In the *Detail* view of the *Results* tab, select one or more result points in the corner columns for which you do not want to run the simulation.



Note: You can cancel the simulation in `pending` as well as `running` state.

2. Right-click on one of the selected points and choose *Cancel Selected Simulation(s)*.

   ADE XL cancels the simulation for the points corresponding to the selected cells and completes the remaining simulations. The status of the simulations that you stopped appears as `canceled`, as shown below.



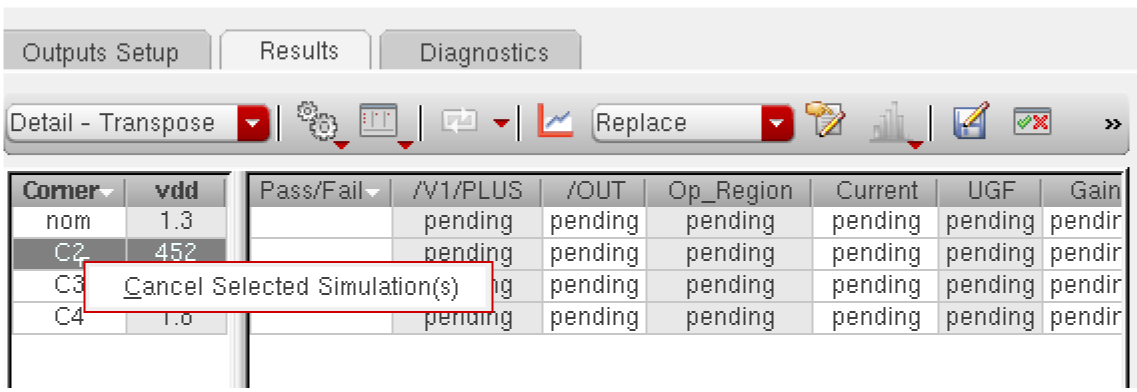
Important Points to Note

- When you cancel the simulations for a corner, ADE XL cancels the simulations only for the tests corresponding to the selected points. For other tests, simulations for those corner and point combinations are completed.

- To cancel simulations for all the corners in a test, hold down the `Shift` key and select all the corners in a row for that test, right-click and choose *Cancel Selected Simulation(s)*. If there are multiple design points, all the corner simulations for only that design point and test combination will be canceled.

- To cancel simulations for a particular corner in all the tests, hold down the `Shift` key and select all the rows in the corner column, right-click and choose *Cancel Selected Simulation(s)*.

- Canceled simulations are considered to be specification failures, so the *Pass/Fail* column shows the `Fail` status for any output specification that contains canceled points.
- When you plot an output across corners, the canceled corners for which results were not generated are not shown in the plot.
- You can cancel simulations for the selected corners in the *Detail - Transpose* view as well. For this, do the following:
  - To cancel simulations for one or more corners across all the tests, in the left pane of the view, click to select the rows for the corners, right-click and choose *Cancel Selected Simulations(s)*, as shown in the figure below.



    Simulations for the corresponding design point and corner combinations are canceled across all the tests.

  - To cancel simulations for specific design point, corner, and test combinations, select the corresponding cells in the right pane of this view, right-click and choose *Cancel Selected Simulations(s)*.

Canceling Simulations for Selected Tests

To cancel simulation for one or more tests, do one of the following:

1. In the *Detail* view of the *Results* tab, select one or more tests in the *Test* column.
2. Right-click on one of the selected test names and choose *Cancel Selected Tests(s)*.

ADE XL cancels all the pending or running simulations for the selected tests and completes the remaining tests.

Important Points to Note

- You can cancel simulations for the selected tests in the *Summary* view as well. For this, select one or more rows for tests, right-click and choose *Cancel Selected Tests(s)*.
- You can also select a combination of test and corners to cancel their simulations. In this case, when you right-click any one of the selected tests or corners, the name of the command changes to *Cancel Selected Test(s) and Simulation(s)*.

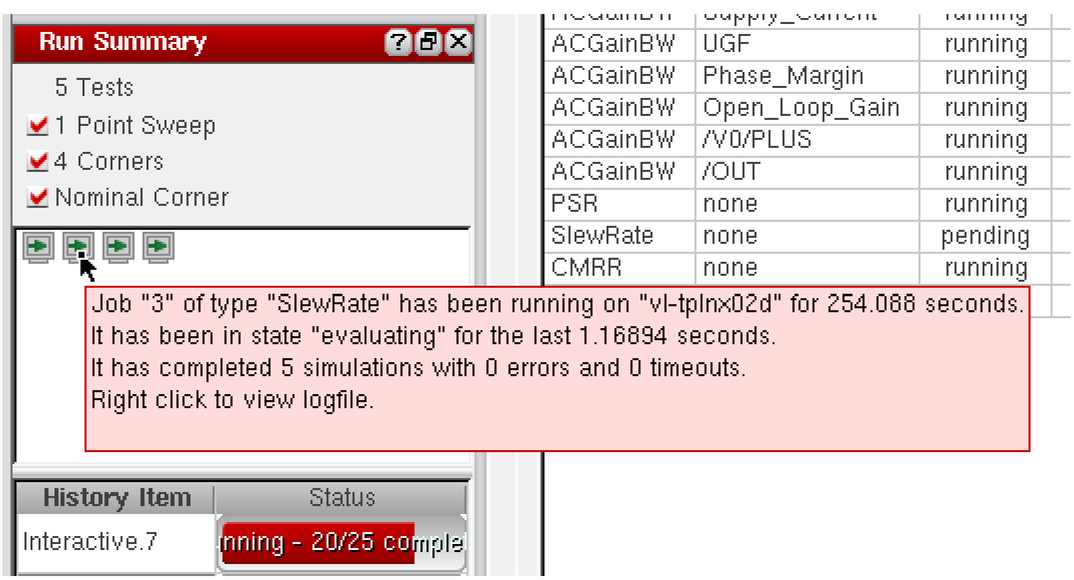
Known Limitations

You cannot cancel selected simulations in the following scenarios:

- While running Monte Carlo Sampling simulations with Auto Stop enabled
- While running simulations with Reliability Analysis

Stopping Jobs and Resubmitting Simulations

During a simulation, the *Run Summary* assistant shows the status of the ICRP jobs that are running simulations. You can check the status of each individual ICRP job by placing the pointer over it. A tooltip is displayed giving the details of the current simulation in progress and the total number of simulations completed by the job. An example is shown in the following figure.



If you observe that a particular job is taking an unusually long time to complete, you can stop that job and resubmit the simulation.

➡ To stop a job and resubmit the simulation, right-click on the job and choose *Stop and Resubmit*.

When you use this command, ADE XL does the following:

- The icon for the stopped job changes to  and then disappears. A new job is created to replace the stopped job.
- The simulation that was running on the stopped job is placed in the beginning of the queue of pending simulations. The simulation for this point is then submitted to the next available job.
- If no more simulations are pending, ADE XL might not create a new job.

Stopping a job is generally helpful in the following scenarios:

- When a job is submitted to a machine that is known to be slow or with limited memory: If the tooltip of a job shows the name of the slow machine and you observe that the currently running simulation is taking an unusually long time to complete, you can stop the job and restart it. The queuing software might distribute the resubmitted job to another machine that has better resources.
- When a particular job has simulated a significantly smaller than the average number of simulations: If the tooltip of a job shows that the number of points simulated by it are less than the points simulated by other jobs, you can stop the slow job and resubmit it.
- When nearing the end of a run, only one or two jobs remain pending and are taking more than the expected time to complete the simulation for the current design points: In this case, you can try to stop and resubmit the job. Resubmission of job can allocate it to another better machine.

Note: If no simulation is running on a job, the context-sensitive menu shows the *Stop* command in place of the *Stop and Resubmit* command.

Ignoring Design Changes During Run

Important

This feature is not supported when the ADE XL setup includes device parameterization.

If you modify the schematic while a simulation run is in progress, the design changes may get considered for the netlist creation and simulation of the pending design points in the current run. This is because a new netlist is created before simulating every design point and any design change, such as, a change in pin name, instance name, instance parameter property, or config binding or any addition or deletion of a design component, may get reflected in the netlist.

However, if you want to ignore any design changes in the simulation run that is already running and to consider them only for the subsequent runs, set the `ignoreDesignChangesDuringRun` flag to `t` as shown below:

```
envSetVal("adexl.simulation" "ignoreDesignChangesDuringRun" 'boolean
t)
```

When this flag is set and you run the simulation, a netlist is created upfront for each test in ADE XL and saved as a master netlist in the top psf directory for the respective test in the simulation directory. A progress bar is also displayed for each test in the ADE XL window to indicate that the netlist is being generated. The saved master netlist for each test is then reused for all the point simulations for the test.

In addition, the user setting for the singleNetlistForAllPoints and includeStatementForNetlistInSimInputFile environment variables are ignored when the *`ignoreDesignChangesDuringRun`* variable is set. That is, the netlist file in the netlist directory for each point is symbolically linked to the master netlist residing in the top level PSF netlist directory that is saved under the top level psf directory for the test. The following figure shows the netlist directory for point `1` of the `Interactive.2` simulation run.

```
Interactive.2:
Interactive.2: cd 1/training:ampTest:1/netlist/
```

```
 48 Nov 18 14:08 .designVariables
 36 Nov 18 14:08 .adc_cascode_opamp.parameters
 53 Nov 18 14:08 netlistHeader -> ../../../psf/training:ampTest:1/netlist/netl
 53 Nov 18 14:08 netlistFooter -> ../../../psf/training:ampTest:1/netlist/netl
 59 Nov 18 14:08 .__master.termorder -> ../../../psf/training:ampTest:1/netlis
 60 Nov 18 14:08 .__master.optionFile -> ../../../psf/training:ampTest:1/netli
 48 Nov 18 14:08 .control -> ../../../psf/training:ampTest:1/netlist/.control*
 55 Nov 18 14:08 .includedModels -> ../../../psf/training:ampTest:1/netlist/.i
 47 Nov 18 14:08 netlist -> ../../../psf/training:ampTest:1/netlist/netlist
 46 Nov 18 14:08 si.env -> ../../../psf/training:ampTest:1/netlist/si.env
 54 Nov 18 14:08 si.foregnd.log -> ../../../psf/training:ampTest:1/netlist/si.
 44 Nov 18 14:08 ihnl -> ../../../psf/training:ampTest:1/netlist/ihnl/
 44 Nov 18 14:08 amap -> ../../../psf/training:ampTest:1/netlist/amap/
 43 Nov 18 14:08 map -> ../../../psf/training:ampTest:1/netlist/map/
114 Nov 18 14:08 artSimEnvLog
173 Nov 18 14:08 designInfo
```

Note that the netlist for point `1` is a symbolic link to the netlist in the `netlist` directory under the psf directory for the test.

The `input.scs` file for Spectre and `input.ckt` file for Ultrasim simulator contain the `include "netlist"` statement, which reflects that the netlist being used is the same netlist for all the points. The following figure shows the `input.scs` file for point `1`.

```
simulator lang=spectre
global 0
parameters IREF=70u SIDDQ=0 VDD=1.8 VIN_CM=1.09
include "gpdk045.scs" section=mc
include "netlist"
simulatorOptions options reltol=1e-3 vabstol=1e-6 iabstol=1e-12 temp=29.0 \
    tnom=27 scalem=1.0 scale=1.0 gmin=1e-12 rforce=1 maxnotes=5 maxwarns=5 \
    digits=5 cols=80 pivrel=1e-3 sensfile="../psf/sens.output" \
    checklimitdest=psf
ac ac freq=100K param=VIN_CM start=1 stop=1G dec=20 annotate=status
dcOp dc write="spectre.dc" maxiters=150 maxsteps=10000 annotate=status
```

This helps in minimizing the size of the simulation directory.

Limitation

When the ADE XL setup includes device parameterization, an incremental netlist needs to be generated for each simulation point during the simulation run. This is required to execute any CDF callbacks during the run. Therefore, any changes that are made to the schematic design during this simulation run cannot be ignored.

In this case, a warning pop-up is displayed in the ADE XL window requiring confirmation that you want to continue. Now, if you continue, any design changes that happen while the simulation is running are considered in the current run.

Continuing the In-Process Simulations After ADE XL GUI Exits

By default, in case of an abrupt exit of ADE XL or Virtuoso GUI, any in-process ADE XL simulation stops immediately. This results in loss of results of the incomplete simulations and there is no way to recover them.

However, you can choose to keep the in-process simulations running in the background even if the ADE XL or Virtuoso GUI is closed. After the simulations are complete, their results are saved in the results database. Next time, you open the ADE XL GUI and if the in-process simulations are complete, you can load the history to view results. In this case, you do not need to rerun the simulations that were running when the ADE XL GUI was closed.

See the following topics for more information:

- Enabling Continuation of the In-Process Simulations on page 475
- Supported Scenarios on page 476
- Limitations on page 482

Enabling Continuation of the In-Process Simulations

To continue and complete the in-process simulations even after the ADE XL GUI is closed, set the *`continueICRPRunOnAbruptGUIExit`* environment variable to t. When the *`continueICRPRunOnAbruptGUIExit`* variable is set to `t`, the IC Remote Processes (ICRPs) that were running the simulations are kept active to complete the simulations. By default, the ICRPs are closed immediately after the ADE XL GUI is exited.

Important

After the ADE XL GUI is closed abruptly, the existing ICRPs will use the ADE XL license to complete the in-process simulations. If more than one ICRP was running, all the processes will share a single license. A new ADE XL session will require another license token.

Supported Scenarios

This feature is supported for the following scenarios:

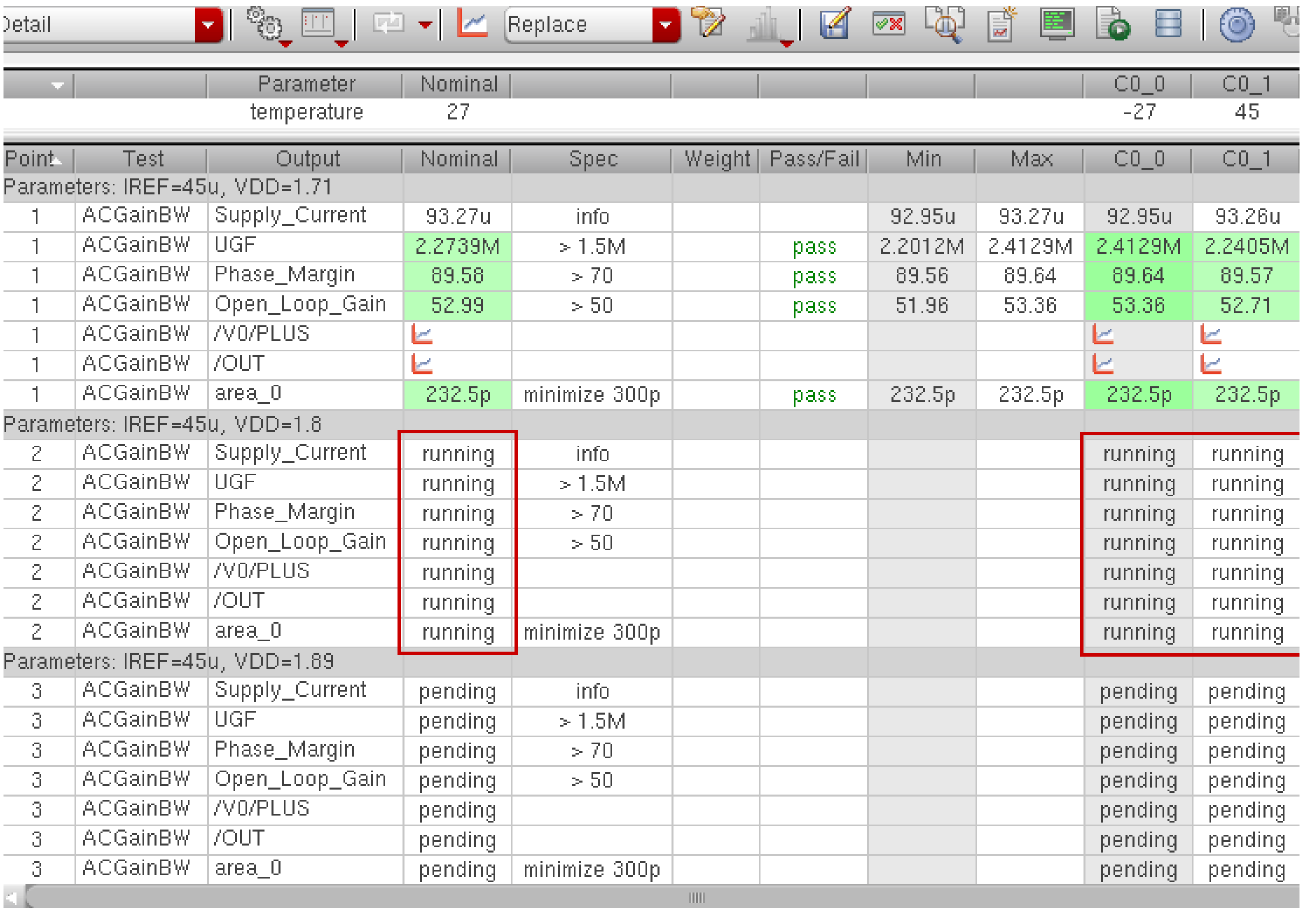
- When the ICRPs are Running and the ADE XL GUI Exits Abruptly
- When the ICRPs are Running and Virtuoso Exits Abruptly
- When the Simulations are Running and Virtuoso is Closed

The following sections describe how the running simulations are completed in the scenarios given above.

When ICRPs are Running and Virtuoso or ADE XL Environment Exits Abruptly

In the first two scenarios, when the Virtuoso or ADE XL GUI exits abruptly, the ICRPs complete the simulations that are currently running, evaluate the output expressions, save results in the results database and then exit.

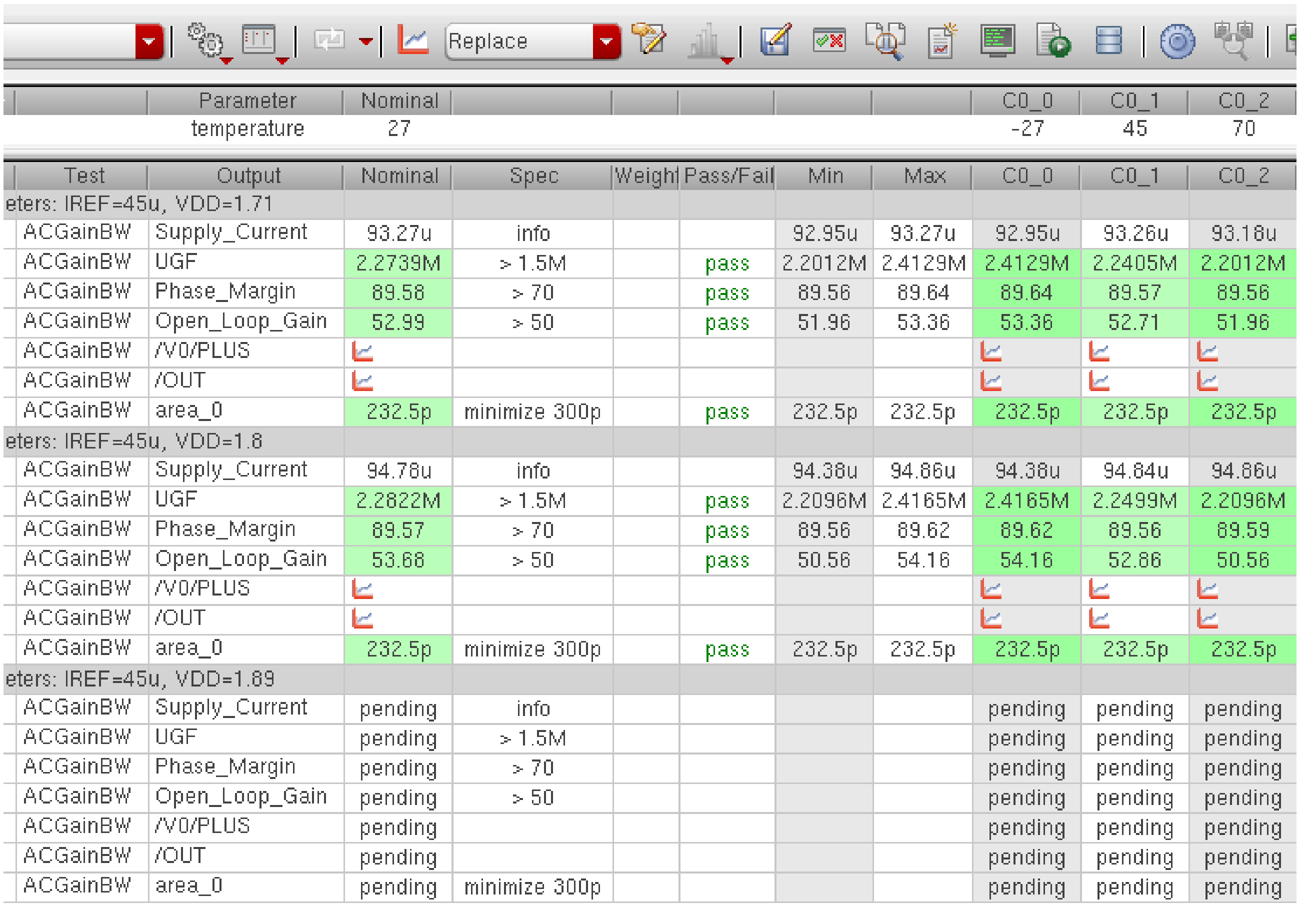
Consider the following scenario in which the ADE XL GUI is closed abruptly when some of the simulations are in the `running` state, as shown below.



	Parameter	Nominal	C0_0	C0_1
	temperature	27	-27	45

Point	Test	Output	Nominal	Spec	Weight	Pass/Fail	Min	Max	C0_0	C0_1
Parameters: IREF=45u, VDD=1.71										
1	ACGainBW	Supply_Current	93.27u	info			92.95u	93.27u	92.95u	93.26u
1	ACGainBW	UGF	2.2739M	> 1.5M		pass	2.2012M	2.4129M	2.4129M	2.2405M
1	ACGainBW	Phase_Margin	89.58	> 70		pass	89.56	89.64	89.64	89.57
1	ACGainBW	Open_Loop_Gain	52.99	> 50		pass	51.96	53.36	53.36	52.71
1	ACGainBW	/V0/PLUS								
1	ACGainBW	/OUT								
1	ACGainBW	area_0	232.5p	minimize 300p		pass	232.5p	232.5p	232.5p	232.5p
Parameters: IREF=45u, VDD=1.8										
2	ACGainBW	Supply_Current	running	info					running	running
2	ACGainBW	UGF	running	> 1.5M					running	running
2	ACGainBW	Phase_Margin	running	> 70					running	running
2	ACGainBW	Open_Loop_Gain	running	> 50					running	running
2	ACGainBW	/V0/PLUS	running						running	running
2	ACGainBW	/OUT	running						running	running
2	ACGainBW	area_0	running	minimize 300p					running	running
Parameters: IREF=45u, VDD=1.89										
3	ACGainBW	Supply_Current	pending	info					pending	pending
3	ACGainBW	UGF	pending	> 1.5M					pending	pending
3	ACGainBW	Phase_Margin	pending	> 70					pending	pending
3	ACGainBW	Open_Loop_Gain	pending	> 50					pending	pending
3	ACGainBW	/V0/PLUS	pending						pending	pending
3	ACGainBW	/OUT	pending						pending	pending
3	ACGainBW	area_0	pending	minimize 300p					pending	pending

If the `continueICRPRunOnAbruptGUIExit` variable is set to `t`, all ICRPs that are currently running these simulations will continue and complete the simulations. Next time when you open the history of this run and view results, the results will appear as shown below.

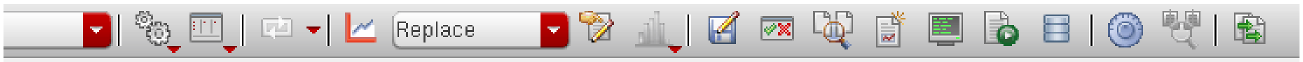


	Parameter	Nominal						C0_0	C0_1	C0_2
	temperature	27						-27	45	70

Test	Output	Nominal	Spec	Weight	Pass/Fail	Min	Max	C0_0	C0_1	C0_2
eters: IREF=45u, VDD=1.71										
ACGainBW	Supply_Current	93.27u	info			92.95u	93.27u	92.95u	93.26u	93.18u
ACGainBW	UGF	2.2739M	> 1.5M		pass	2.2012M	2.4129M	2.4129M	2.2405M	2.2012M
ACGainBW	Phase_Margin	89.58	> 70		pass	89.56	89.64	89.64	89.57	89.56
ACGainBW	Open_Loop_Gain	52.99	> 50		pass	51.96	53.36	53.36	52.71	51.96
ACGainBW	/V0/PLUS									
ACGainBW	/OUT									
ACGainBW	area_0	232.5p	minimize 300p		pass	232.5p	232.5p	232.5p	232.5p	232.5p
eters: IREF=45u, VDD=1.8										
ACGainBW	Supply_Current	94.78u	info			94.38u	94.86u	94.38u	94.84u	94.86u
ACGainBW	UGF	2.2822M	> 1.5M		pass	2.2096M	2.4165M	2.4165M	2.2499M	2.2096M
ACGainBW	Phase_Margin	89.57	> 70		pass	89.56	89.62	89.62	89.56	89.59
ACGainBW	Open_Loop_Gain	53.68	> 50		pass	50.56	54.16	54.16	52.86	50.56
ACGainBW	/V0/PLUS									
ACGainBW	/OUT									
ACGainBW	area_0	232.5p	minimize 300p		pass	232.5p	232.5p	232.5p	232.5p	232.5p
eters: IREF=45u, VDD=1.89										
ACGainBW	Supply_Current	pending	info					pending	pending	pending
ACGainBW	UGF	pending	> 1.5M					pending	pending	pending
ACGainBW	Phase_Margin	pending	> 70					pending	pending	pending
ACGainBW	Open_Loop_Gain	pending	> 50					pending	pending	pending
ACGainBW	/V0/PLUS	pending						pending	pending	pending
ACGainBW	/OUT	pending						pending	pending	pending
ACGainBW	area_0	pending	minimize 300p					pending	pending	pending

Note that the results of the simulations were completed and saved in the results database. You need to run only the pending simulations, whose status appears as `pending`. For this, on the *History* tab of the *Data View* assistant pane, right-click on the history item that was running when the GUI exited and choose *Re-run Unfinished/Error Points*.

If the *`continueICRPRunOnAbruptGUIExit`* variable is set to `nil`, all the ICRPs are immediately exited without completing the simulations that were already running. In such a case, if you open the history results, the status of simulations is shown as `running`, but actually they are not running.



	Parameter	Nominal					C0_0	C0_1	C0_2
	temperature	27					-27	45	70

Test	Output	Nominal	Spec	Weight	Pass/Fail	Min	Max	C0_0	C0_1	C0_2
IREF=45u, VDD=1.71										
GainBW	Supply_Current	93.27u	info			92.95u	93.27u	92.95u	93.26u	93.18u
GainBW	UGF	2.2739M	> 1.5M		pass	2.2012M	2.4129M	2.4129M	2.2405M	2.2012M
GainBW	Phase_Margin	89.58	> 70		pass	89.56	89.64	89.64	89.57	89.56
GainBW	Open_Loop_Gain	52.99	> 50		pass	51.96	53.36	53.36	52.71	51.96
GainBW	/V0/PLUS									
GainBW	/OUT									
GainBW	area_0	232.5p	minimize 300p		pass	232.5p	232.5p	232.5p	232.5p	232.5p
IREF=45u, VDD=1.8										
GainBW	Supply_Current	94.78u	info			94.38u	94.78u	94.38u	running	running
GainBW	UGF	2.2822M	> 1.5M			2.2822M	2.4165M	2.4165M	running	running
GainBW	Phase_Margin	89.57	> 70			89.57	89.62	89.62	running	running
GainBW	Open_Loop_Gain	53.68	> 50			53.68	54.16	54.16	running	running
GainBW	/V0/PLUS								running	running
GainBW	/OUT								running	running
GainBW	area_0	232.5p	minimize 300p			232.5p	232.5p	232.5p	running	running
IREF=45u, VDD=1.89										
GainBW	Supply_Current	running	info					running	pending	pending
GainBW	UGF	running	> 1.5M					running	pending	pending
GainBW	Phase_Margin	running	> 70					running	pending	pending
GainBW	Open_Loop_Gain	running	> 50					running	pending	pending
GainBW	/V0/PLUS	running						running	pending	pending
GainBW	/OUT	running						running	pending	pending
GainBW	area_0	running	minimize 300p					running	pending	pending

If you rerun the unfinished points, all the simulations that were in the `running` or `pending` state are run again.

Important Points to Note

- If you reopen Virtuoso and load the history results while the simulations are still running, you might not see results of the in-process simulations that have the status set as `running`. In such cases, it is recommended to view the results again after an estimated time of simulation run.

  Note: The ADE XL GUI of the reopened Virtuoso session does not provide any status update about the ICRP jobs started by the previous Virtuoso session, but you can use the job monitoring commands provided by your DRMS to monitor the status of the jobs. For example, you can use the `bjobs` command to monitor the LSF jobs. After verifying that all the ICRP jobs started by the previous Virtuoso session have completed, you can expect to view the results by reloading the history, as shown below.

a. On the *Results* tab, make sure to close any opened instance of the history that was running when the GUI exited.

b. On the *History* tab of the Data View assistant pane, right-click on the history item that was running when the GUI exited and choose *Load Setup to Active* or *View Results*.

- If you reopen Virtuoso and ADE XL immediately after the GUI was closed abruptly and the in-process simulations are still running, you might not see the updated results after reloading the results. The results are available only after the simulations are complete and saved to the results database.

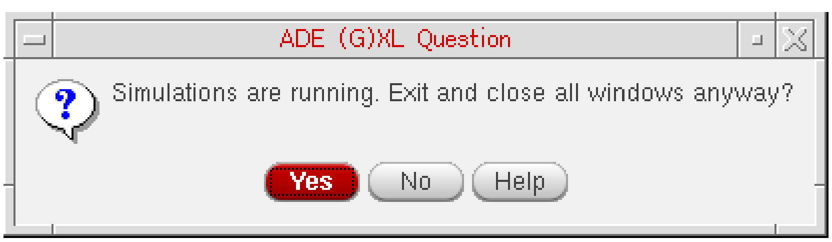
  To view the results:

  a. On the *Results* tab, make sure to close any opened instance of the history that was running when the GUI exited.

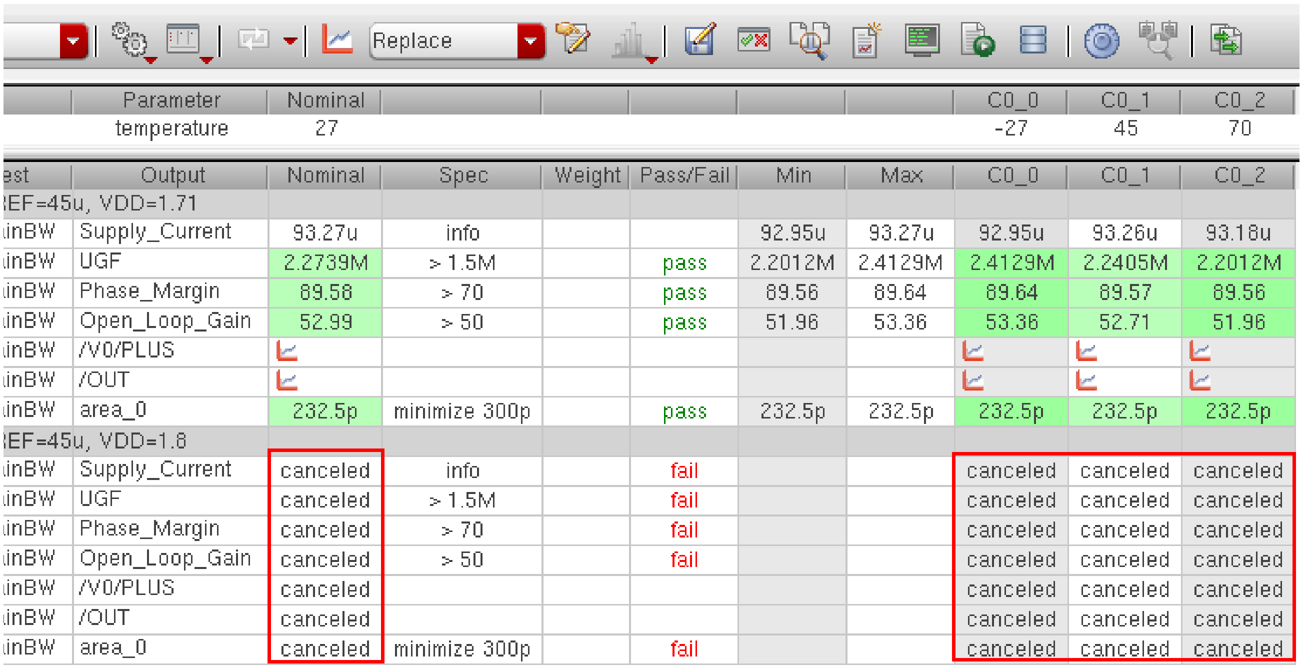
  b. On the *History* tab of the *Data View* assistant pane, right-click on the history item that was running when the GUI exited and choose *Load Setup to Active* or *View Results*.

When the Simulations are Running and Virtuoso is Closed

By default, if the `continueICRPRunOnAbruptGUIExit` variable is set to `nil` and you close Virtuoso while simulations are running, Virtuoso displays the following message to alert that some simulations are running and to confirm if you want to exit Virtuoso without completing those simulations.



If you click *Yes*, all running simulations are stopped and Virtuoso is closed. If you click *Yes*, simulations that are currently running are stopped and ADE XL is closed. Next time, when you view the history results, the status of the pending simulations is shown as `Canceled`, as shown below.



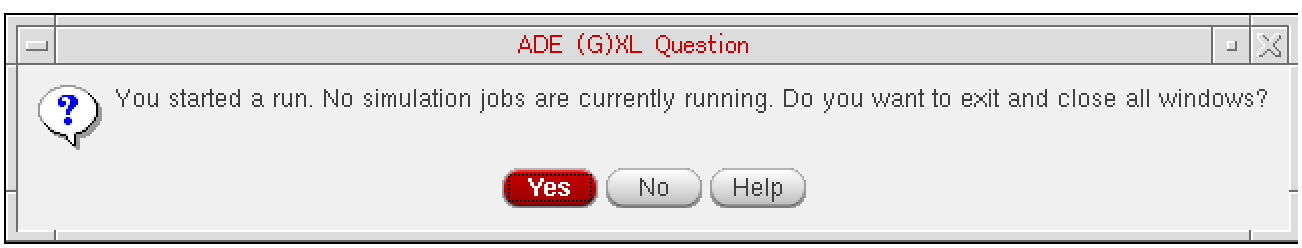
	Parameter	Nominal						C0_0	C0_1	C0_2
	temperature	27						-27	45	70

est	Output	Nominal	Spec	Weight	Pass/Fail	Min	Max	C0_0	C0_1	C0_2
EF=45u, VDD=1.71										
inBW	Supply_Current	93.27u	info			92.95u	93.27u	92.95u	93.26u	93.18u
inBW	UGF	2.2739M	> 1.5M		pass	2.2012M	2.4129M	2.4129M	2.2405M	2.2012M
inBW	Phase_Margin	89.58	> 70		pass	89.56	89.64	89.64	89.57	89.56
inBW	Open_Loop_Gain	52.99	> 50		pass	51.96	53.36	53.36	52.71	51.96
inBW	/V0/PLUS									
inBW	/OUT									
inBW	area_0	232.5p	minimize 300p		pass	232.5p	232.5p	232.5p	232.5p	232.5p
EF=45u, VDD=1.8										
inBW	Supply_Current	canceled	info		fail			canceled	canceled	canceled
inBW	UGF	canceled	> 1.5M		fail			canceled	canceled	canceled
inBW	Phase_Margin	canceled	> 70		fail			canceled	canceled	canceled
inBW	Open_Loop_Gain	canceled	> 50		fail			canceled	canceled	canceled
inBW	/V0/PLUS	canceled						canceled	canceled	canceled
inBW	/OUT	canceled						canceled	canceled	canceled
inBW	area_0	canceled	minimize 300p		fail			canceled	canceled	canceled

If you want that even if you close Virtuoso, the simulations that are already running should continue to run, set the `continueICRPRunOnAbruptGUIExit` variable is set to `t`. If this variable is set, when you close Virtuoso, it checks the state of simulation runs. There can be the following two cases:

- Simulation run has started, but no job is running when you close Virtuoso

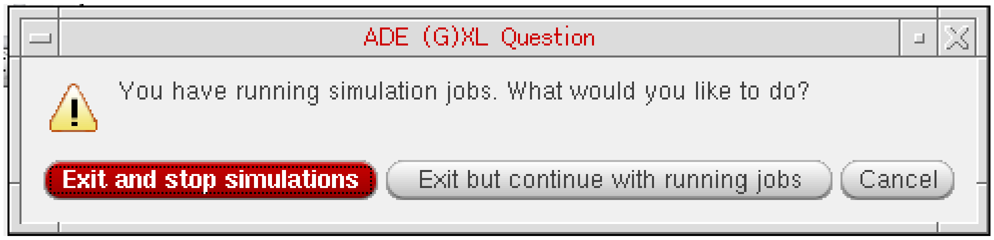
  In this case, the following message is displayed to alert that you started some simulation runs that are pending to run.



  To exit Virtuoso and cancel all the pending simulations, click *Yes*. Click *No* to stop from exiting Virtuoso.

- Simulation run has started and one or more jobs are running

  In this case, the following message is displayed to confirm if you want to continue the running simulations or stop them after closing Virtuoso.



To exit Virtuoso without completing the running simulation jobs, click *Exit and stop simulations*.

To exit Virtuoso, but to continue running simulations, click *Exit but continue with running jobs*. Next time, when you run Virtuoso, you can load the history results to view the updated simulation results.

Limitations

The known limitations of this feature are as follows:

- While the ICRPs of a history are running to complete the in-process simulations, if you restart the ADE XL GUI, the new session is not connected to the already running ICRPs.
- A new ADE XL session will not share a license with the ICRPs that are already running to complete simulations for a history run. The new ADE XL session will require another ADE XL license.
- If you open a new ADE XL session while the in-process simulations of a history are still running and load the results of that history, the status of the in-process runs appears as `running`. After the simulations are complete, the status is not refreshed automatically. You need to reload the history to view the updated status.

Viewing Job Status

For each job the program submits, a computer terminal icon appears on the Run Summary assistant pane. Each icon indicates the status of its respective job as follows:

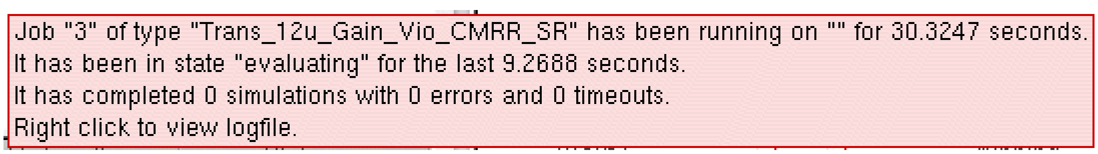
Icon	Job Status
	Pending
	Starting
	Configuring

Icon	Job Status
	Running
	Idle
	Cleaning up
	Completed

Additionally, you can monitor the job status by doing the following:

➤ Hover the mouse cursor over the job status icon.

A job status pop-up appears.


Job "3" of type "Trans_12u_Gain_Vio_CMRR_SR" has been running on "" for 30.3247 seconds.
It has been in state "evaluating" for the last 9.2688 seconds.
It has completed 0 simulations with 0 errors and 0 timeouts.
Right click to view logfile.

The job status pop-up contains information about how long the job has been running, what state it is in, how many simulations have completed with how many errors and how many timeouts.

To view the log file while the program is writing it, do the following:

➤ Right-click the job status icon.

The log file appears in a separate window. Simulator messages appear in the window as the program writes them. As the window fills up with messages, the program scrolls down so that the most recent messages are always visible.

Viewing the Netlist

To view the netlist file, do the following:

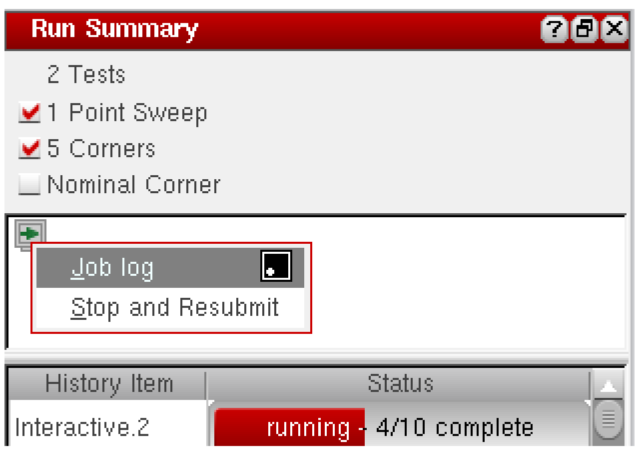
➤ On the Results tab of the Outputs pane:

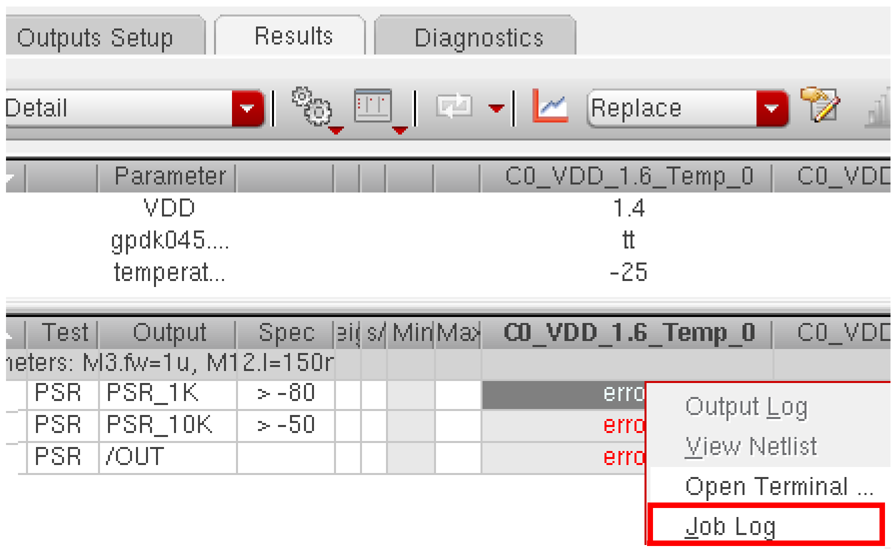
- ❑ Right-click a test or the result for the nominal corner and choose *View Netlist* to view the netlist for the nominal corner.
- ❑ Right-click the result for a corner and choose *View Netlist* to view the netlist for the corner.

The netlist file appears in a text window.

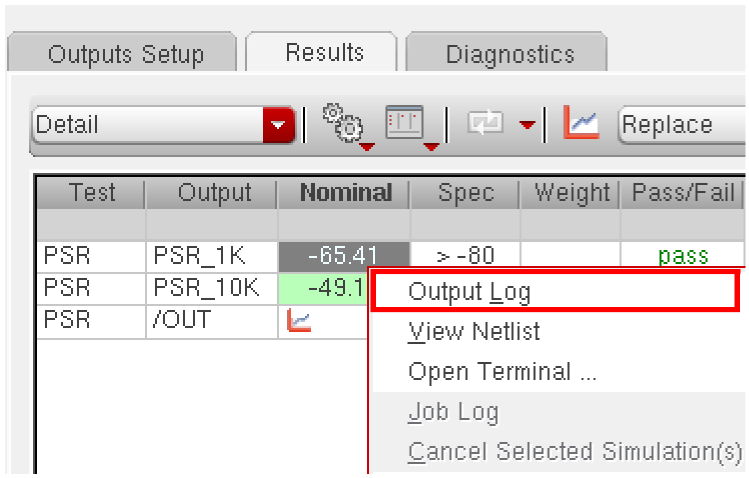
Viewing the ADE XL Logs After Running Simulations

After running simulations, ADE XL saves the information of ICRP jobs, simulation runs, and outputs in different log files. This section describes the various types of log files ADE XL maintains and the details provided in those files.

Log File	Contains...
Job log (for an ICRP job)	Information about an ICRP job and all the simulations that were run on that job. It includes various details, such as the Cadence software version number, operating system version, working directory, the testbench details, variables and parameters for every run, simulator details, results location, data directory, netlist directory, and simulation errors (if any). Each ICRP has a separate log file. To view the job log for an ICRP, right-click an ICRP icon in the *Run Summary* assistant and choose *Job log*.  Also see: Viewing the Job Log

Log File	Contains...
Job log (for a data point)	Information related to the job for a selected data point. This log contains the details for a data point such as the variable and parameter values, the path to the netlist directory and the data directory, and the output or error details.To view the output log for a point, right-click on any result value for that point on the *Results* tab and choose *Job Log*. The simulator log for that point is displayed in a new window.  Environment variable to control the creation of the job log for a data point: generateJobFileOnlyOnError At the data point level, ADE XL saves job logs only for the simulations that result into an error. To save the job logs for all the jobs irrespective of their result status, set the generateJobFileOnlyOnError environment variable to `nil`. However, note that this would increase the disk space requirement.

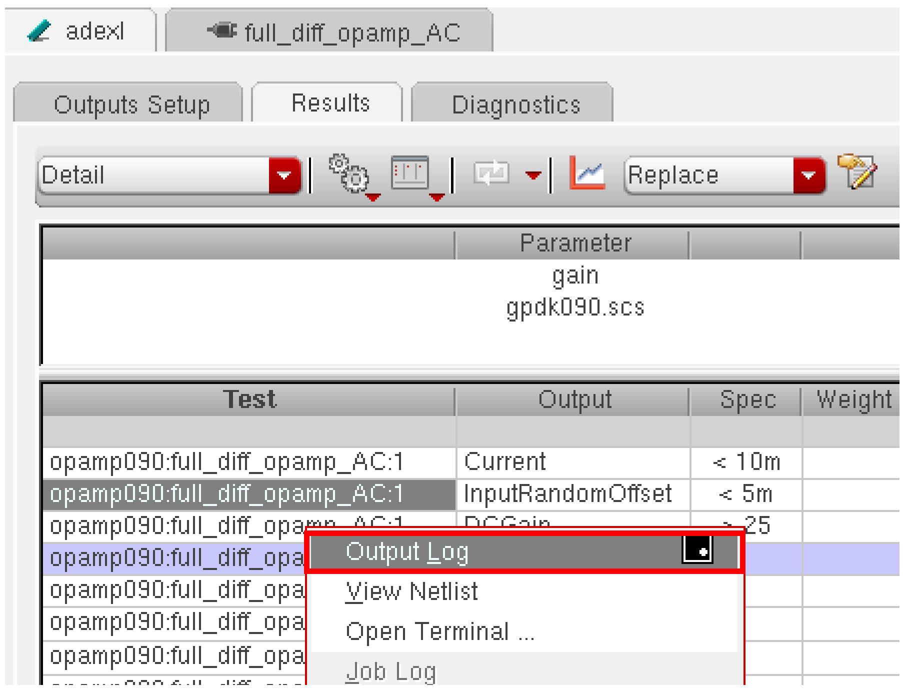
Log File	**Contains...**
Output log	The simulator output log for a particular data point. To view the output log for a point, right-click on any result value for that point on the *Results* tab and choose *Output Log*. The simulator log for that point is displayed in a new window.



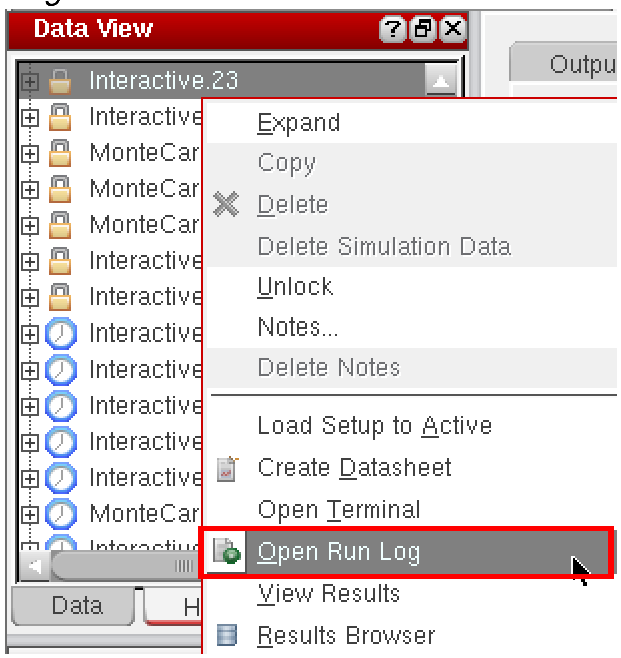
Log File	Contains...
Composite output log	The composite output logs for all the simulations run for all the points and corners. In this composite log, log for each simulation is separated by the following two statements:

```
********* LOG ENDS **************
**                             **
**                             **
********* LOG STARTS ************
```

To view the composite output log, on the *Results* tab, right-click in the *Test* or *Output* column and choose *Output Log*.



Environment variable to control the creation of the composite output log depending on the number of simulations: createCompositeSimLogFileWhenSimCountFewerThan

Log File	Contains...
Run log	The summary of a simulation run in ADE XL. To view the run log, on the *History* tab of the *Data View* assistant, right-click on a history item and choose *Open Run Log*.  This log contains the start time and end time for the simulation run, the number of points completed and the number of simulation errors. In addition to these, by default, ADE XL also saves the details of the best design point when the number of points is less than the count specified by createRunLogWhenSimsFewerThan or when the Single Run, Sweeps and Corners run mode is run as part of the Manual Tuning run mode. You can choose to include this information in other scenarios as well by using the following two environment variables: Environment variables to control the creation of the run job log: ■ createRunLogForSweepsCorners ■ createRunLogWhenSimsFewerThan

Viewing the Job Log

A job log can be viewed for a particular ICRP on which simulation is running or a particular data point in the Results tab.

Viewing Job Log for an ICRP

To view the log file for a particular ICRP job, do the following:

➡ In the Run Summary assistant pane, right-click the job status icon  and click *Job Log*.

The Job.log file appears in a text window. This file contains information about the ICRP job that is running simulations. It includes various details, such as the Cadence software version number, operating system version, working directory, design path, simulator, results location, data directory, netlist directory, and simulation errors, if any.

How Virtuoso Saves the Job Logs?

By default, each Virtuoso process creates a unique log subdirectory such as logs<num> under logs_<username>, where all the ICRP processes started by that Virtuoso process write their job log files. For example, if you start a Virtuoso process from a directory and that process further starts three ICRP jobs, Virtuoso creates a directory logs_<username>/logs0 and the three ICRPs create their job log files, Job0.log, Job1.log and Job2.log in logs_<username>/logs0 directory. While this Virtuoso process is running, if the user starts another Virtuoso process from the same directory, which further starts two ICRP jobs, the second Virtuoso process will create a directory logs_<username>/logs1 and two ICRPs started by it write their job log files Job0.log and Job1.log in the logs_<username>/logs1 directory.

Viewing Job Log for a Result

Note: By default, ADE XL saves job logs only for the simulations that result into an error.

To view the job log file for a particular data point in the Results tab, do the following:

➡ On the Results tab of the Outputs pane, right-click a result cell and choose *Job Log*.

The job log file for the selected result appears in a text window. This file contains all the information related to the job, such as the parameter values, the path to the netlist directory and the data directory, and the output or error details.

To view the job logs for multiple points in the same log window, do the following:

➡ On the Results tab of the Outputs pane, hold down the *Ctrl* key and click multiple cells, right-click and choose *Job Log*.

Important Points to Note

- ADE XL saves the job log files in the psf directory and can be viewed when you load the results for a history.
- By default, to enable debugging, ADE XL saves the job log files only for those jobs that failed due to some reason and did not give any result. The *Job Log* command is not enabled for the points that successfully showed results. To save the job logs for all the jobs, set the generateJobFileOnlyOnError environment variable to `nil`. However, note that this would increase the disk space requirement.

Viewing the Simulation Output Log File

To view the simulation log file, do the following:

➤ On the Results tab of the Outputs pane:

- Right-click a test or the result for the nominal corner and choose *Output Log* to view the simulation log file for the nominal corner.
- Right-click the result for a corner and choose *Output Log* to view the simulation log file for the corner.

The simulation output log file appears in a text window. This file contains information about the simulation environment, the command line sent to the simulator, and any simulation error messages.

See also:

Simulation Errors

Simulation Errors

If the simulation of an analysis or evaluation of an expression fails, ADE XL reports appropriate errors in the Results tab or in the output log. The errors can be categorized as described in the following table.

Type of error	Reported when...
netlist error *(error)*	■ Generation of netlist fails because of some reason, such as a missing device master.
simulation error *(sim err)*	■ A simulation was killed. This can be due to various reasons such as process killed in the background or insufficient disk space. ■ The analysis on which an expression is dependent failed due to errors. In this case, all the expressions that depend on the failed analysis show *sim err* in the results table. Other outputs are not affected due to this failure. For more details, refer to evalOutputsOnSimFailure.

Type of error	Reported when...
evaluation error *(eval err)*	■ An expression was wrongly constructed, and therefore, could not be properly evaluated ■ No data was generated after simulation due to some reason ■ The analysis on which evaluation of an expression is dependent was not selected for simulation

Important Points to Note

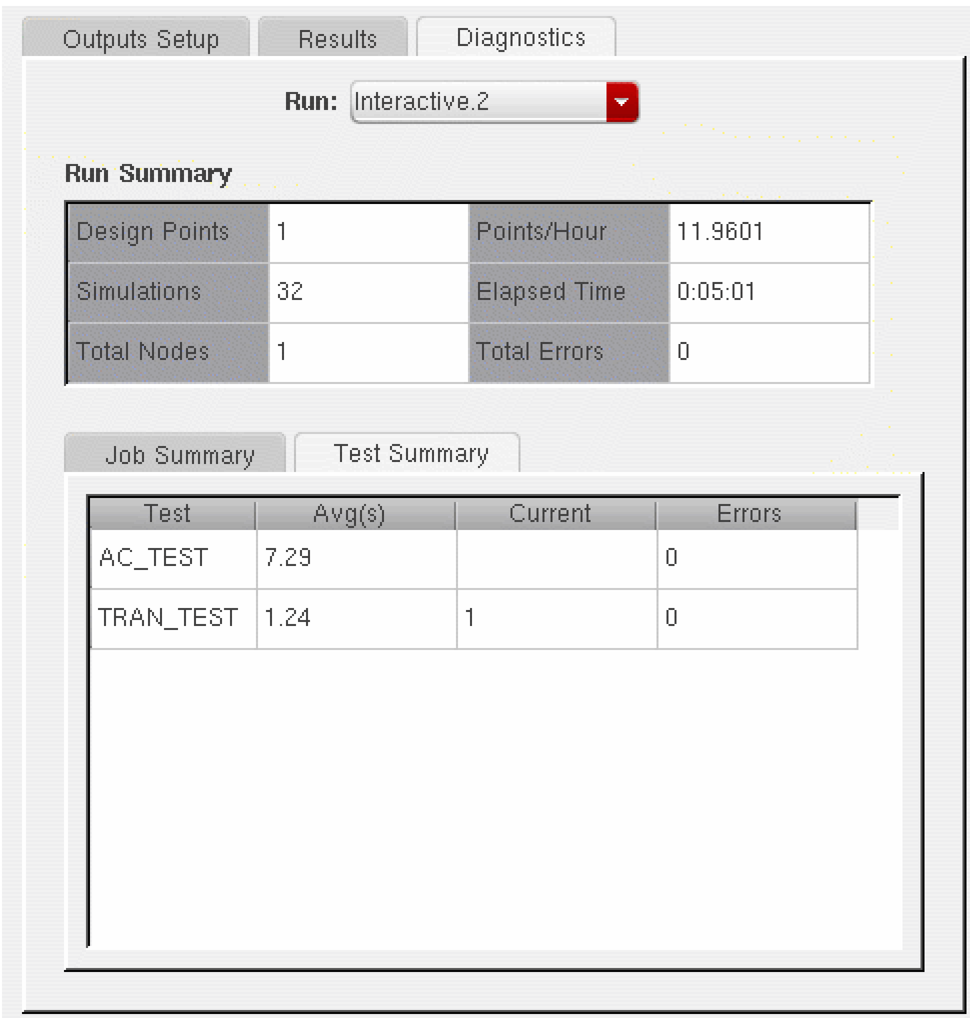
- You can rerun the design points that could not run because of simulation errors. However, you cannot rerun the design points that failed with eval errors because they will result in the same error again.
- If the eval error occurs because of an incorrectly built expression, you can update the expression on the *Output Setup* tab and click *Re-evaluate* on the *Results* tab to rerun the expression evaluation.

See also:

- Viewing the Simulation Output Log File
- Viewing the Job Log
- Simulating Only Error or Incomplete Points
- showOutputLogOnError
- evalOutputsOnSimFailure

Viewing Diagnostics Information

The Diagnostics tab of the Outputs pane displays information on machine utilization, the average time for a simulation run, and other runtime information including the total elapsed time of the simulation run, the number of points run thus far, the total number of nodes in use, and so on.



The Diagnostics tab consists of three parts:

- <u>Run Summary</u> on page 495
- <u>Job Summary</u> on page 495
- <u>Test Summary</u> on page 495

Run Summary

The Run Summary table summarizes the following information about the simulation run.

Field	Description
Design Points	Number of points in the run
Simulations	Total number of simulations
Total Nodes	Total number of simulation nodes used
Points/Hour	Points simulated per hour
Elapsed Time	Elapsed time of the simulation run
Total Errors	Total number of runtime errors for all tests. Runtime errors include netlisting errors, simulation errors, and errors other than evaluation errors.

Job Summary

The Job Summary tab displays the following job related information when a simulation run is in progress.

Field	Description
Job	Job name
Status	Status of the job
Machine	System on which job is running
Test Name	Name of the test for which simulation is running

Test Summary

The Test Summary tab summarizes the following information about the simulation run.

Field	Description
Test	Name of the test

Field	Description
Avg(s)	Average time in seconds the simulation takes to run
Current	Number of simulation nodes that are currently running
Errors	Total number of runtime errors for each test. Runtime errors include netlisting errors, simulation errors, and errors other than evaluation errors.

You can view the information in the Diagnostics tab during and after a simulation run. You can also view the diagnostics information for all the runs or for a specific run.

Note: The Diagnostics tab displays information only for the runs that occurred in the current ADE XL session.

To view the combined diagnostics information for all the runs, do the following in the Diagnostics tab:

➤ In the *Run* drop-down list, select *All Runs*.

To view the diagnostics information for a specific run, do the following in the Diagnostics tab:

➤ In the *Run* drop-down list, select the name of the run for which you want to view the diagnostics information.

To view the log file for a job, do the following:

➤ Right-click on the row for the job in the Job Summary tab.

The `Job.log` file appears in a text window. This file contains information about the job such as Cadence software version number, operating system version, working directory, design path, simulator, results location, data directory, netlist directory, and whether there were any simulation errors. If there were simulation errors, you might be directed to look in the simulation output log file for more information (see “Viewing the Simulation Output Log File”).

Running an Incremental Simulation

The results for simulation runs in ADE XL and ADE GXL are stored in history items. Incremental simulation allows you to specify a reference history item from which you can do either or both of the following:

- Reuse the results for subsequent simulation runs

  After running a simulation, you can add more corners or sweep points or modify the values of the existing ones and see the simulation results of the changed corners or sweep points. For example, you change the value of a variable, say, `sLoad` from `1,2,3` to `1,2,5`. In such a case, you can specify the history item for the previous run as the reference history for the incremental simulation run. This allows you to save time because the results for the completed points (when `sLoad=1` and `sLoad=2`) will be reused from the reference history and simulations will be run only for the change (from `sload=3` to `sload=5`) in the variable value.

  This is also helpful when you have to correct any value of a corner or sweep point. ADE XL will re-run the simulation only for the new corner or sweep point. The results of the matched corners or points will be copied from the reference history to the new history and the undesired points will be filtered out.

- Reuse the netlist for subsequent simulation runs

  For example, if your design has not changed, you can reuse the netlist in the history item for the previous run. You can also reuse the netlist if your design has changed, but you want to run incremental simulation using the netlist for the reference history for debugging that netlist across different operating conditions.

For more information, see the following topics:

- Requirements for Running Incremental Simulation
- Running an Incremental Simulation
- Viewing Differences Between the Active Setup and the Reference History Setup
- Rerunning Simulation after Modifying the Netlist

Requirements for Running Incremental Simulation

Note the following requirements for running incremental simulation:

- You can run incremental simulation only for the following run modes:
  - Single Run, Sweeps and Corners

- Monte Carlo Sampling
- Global Optimization
- Local Optimization

For more information about run modes, see Specifying the Run Mode on page 71.

- The results and the netlist can be reused only from a history item that meets the following requirements:
  - At least one enabled test in the reference history has the same name as an enabled test in the active setup, and that test has not been modified in the active setup.

    A test is considered modified if there is any difference between the analyses (including differences in analysis settings such as change in start time, stop time, and so on), outputs, model setup, and so on for a test in the reference history and the same test in the active setup.

    Note: During incremental simulation, tests that are modified in the active setup will be resimulated.
  - At least one enabled corner (including the nominal corner) in the reference history has the same name as an enabled corner in the active setup, and that corner has not been modified in the active setup.

    A corner is considered modified if:
    - Anything other than the value of variables (including temperature) and parameters is different between a corner in the reference history and the same corner in the active setup. For example, a corner is considered modified if its model setup is different in the reference history and the active setup.
    - Any variable or parameter does not have at least one value that is the same in both the reference history and the active setup.
  - The same set of global variables and parameters exist in the reference history and the active setup.

    Note the following:
    - The global variables and parameters in the active setup can have values that are different from the values in the reference history, but must have at least one value that is the same in both the reference history and the active setup.
    - This condition is not required to be met if only the netlist is reused from a history item.

- ❑ If the current run mode is *Single Run, Sweeps and Corners*, *Local Optimization* or *Global Optimization*, the reference history must also be for one of these run modes.

  For more information about run modes, see Specifying the Run Mode. For information about the name of the history item for various run modes, see Viewing Checkpoints.

- ❑ If the current run mode is Monte Carlo Sampling, the reference history must also be for the Monte Carlo Sampling run mode.

- ■ If the design has changed and the active setup includes device parameterization, incremental simulation cannot be done because the netlist must be recreated for each simulation when device parameters are present. In this case, you must revert your schematic to the version used in the earlier simulation run.

Running an Incremental Simulation

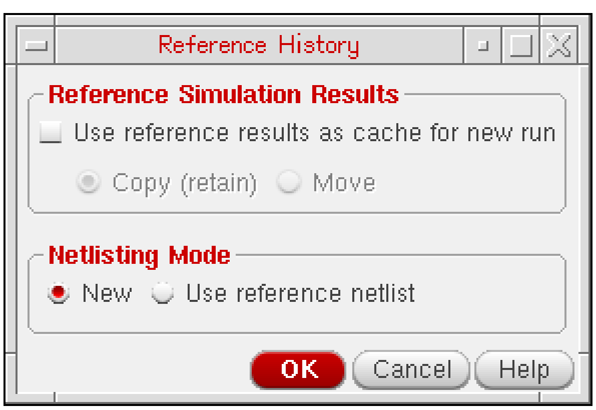
Note: The incremental simulation run uses the active setup. The setup in the reference history is ignored during the run.

To run incremental simulation, do the following:

1. Ensure that the set up of the previous run that had errors is loaded as an active setup in the ADE XL environment.
2. Select the run mode in the *Select a Run Mode* drop-down list on the Run toolbar.

   For more information about run modes, see Specifying the Run Mode on page 71.
3. Click the *Reference History Options* button on the *Reference History* toolbar.

   The Reference History form appears.



4. Select the *Use reference results as cache for new run* check box if you want the results in the reference history to be reused for the incremental simulation run.

   Then do one of the following:

Select	To
Copy (retain)	Copy the simulation results of the reference history to the new history item that is created during the incremental simulation run.
Move	Move the simulation results of the reference history to the new history item that is created during the incremental simulation run. **Note:** If the simulation results are moved, you will not be able to perform post-processing operations (like plotting, printing, annotation, re-evaluation, and so on) on the history item specified as the reference history.

   Note: The results for completed points will be reused from the reference history only for tests that are not modified in the active setup. This is because modified tests will be resimulated during the incremental simulation run.

5. Specify the netlisting mode for the run by doing one of the following:

Select	To
New	Renetlist designs during the incremental simulation run.

Use reference netlist

Reuse the netlist of the reference history for the incremental simulation run. This ensures that all the simulations in the completed history use the same netlist.

Select this option if:

- Your design has not changed after the simulation run in which the reference history (you want to use for the incremental simulation run) was created.
- Your design has changed after the simulation run in which the reference history was created, but you want to run incremental simulation using the netlist for the reference history for debugging that netlist across different operating conditions.

6. Click *OK* to save the changes and close the Reference History form.

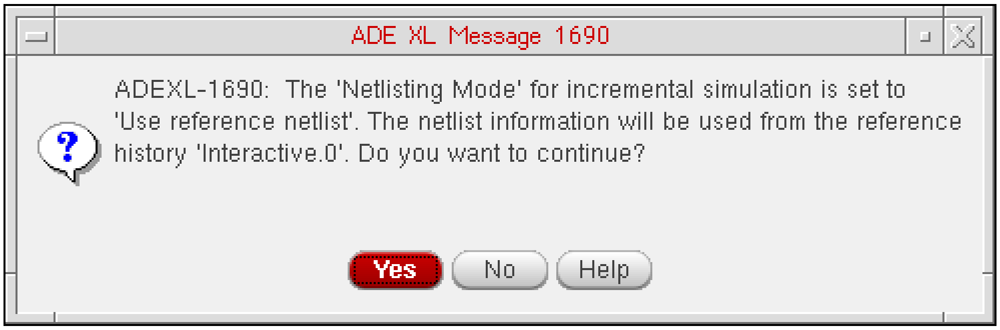
7. In the *Reference* drop-down list on the Reference History toolbar, choose the history item you want to use as the reference history for the incremental simulation run.

   Note: The *Reference* drop-down list displays only the history items from which results and the netlist can be reused during incremental simulation runs. For more information, see Requirements for Running Incremental Simulation on page 497.

8. (Optional) Click the *Show Differences*  button on the Reference History toolbar to view the differences between the active setup and the setup in the reference history. For more information, see Viewing Differences Between the Active Setup and the Reference History Setup on page 502.

9. On the Run toolbar, click the *Run Simulation*  button to start the incremental simulation run.

   The following message box appears if *Use reference netlist* is set as the netlisting mode in the Reference History form. Click *Yes* to continue with the incremental simulation run.



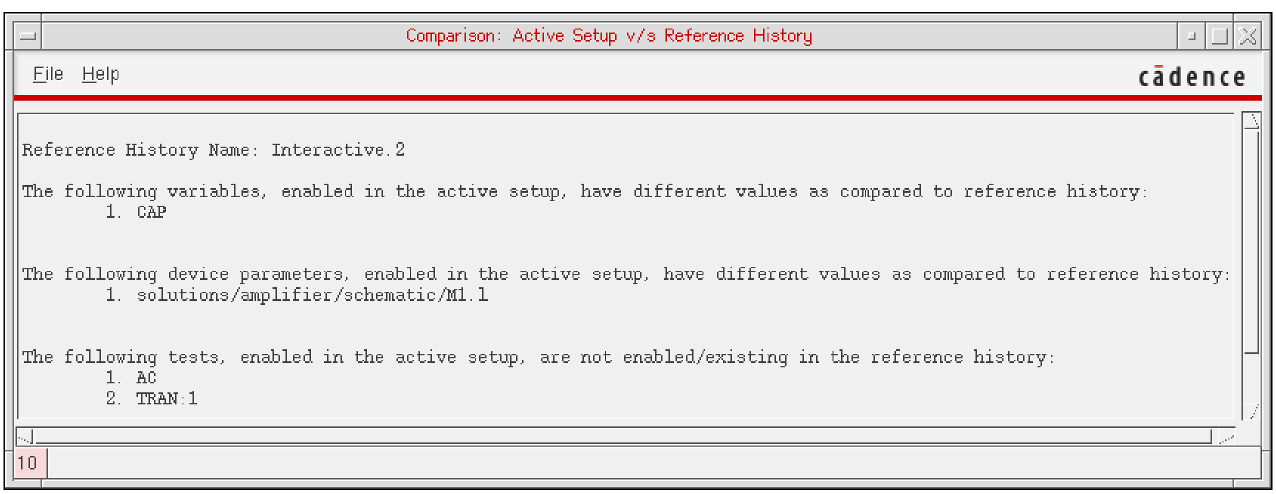
Note that the number of completed points displayed on the run status quickly reflects the number of points already run in the referenced history and for which the results are already available. Result data for those points is reused (copied or moved) from the results of the referenced history. Simulation is then continued for the remaining points that earlier showed errors or were incomplete.

Viewing Differences Between the Active Setup and the Reference History Setup

To view the differences between the active setup and the setup in the reference history, do the following:

➡ Click the *Show Differences*  button on the Reference History toolbar.

The differences between the active setup and the setup in the reference history are displayed in the Comparison: Active Setup v/s Reference History form.



Rerunning Simulation after Modifying the Netlist

After running a simulation, you might need to make some manual modifications in the netlist and re-run a simulation without regenerating a new netlist using ADE XL. In some cases, you might need to completely replace the netlist file with another one to be used to run a simulation with the existing ADE XL setup.

In the scenarios described above, you can perform the following steps to modify or replace a netlist file used in the previous run and use it for the next simulation:

1. After the reference simulation run is complete, note the name of a history saved for it. For example, `Interactive.3`.

2. At the command prompt, traverse to the directory where the netlist was saved for that history.

   Note: By default, the netlist file is saved in the simulation results directory specified by the saveResDir or saveDir environment variables in the given order of preference.

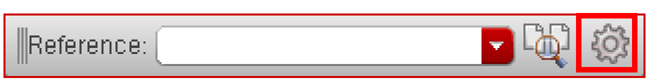
   For example, if the name of the history item for the test `ACGainBW` is `Interactive.14`, the netlist file for this history is saved in the *`<sim-dir-path>`*/simulation/*`<libName>`*/*`<cellName>`*/adexl/results/data/Interactive.14/psf/ACGainBW directory.

3. Open the netlist file in a text editor and make changes as required.

4. Save and close the netlist file.

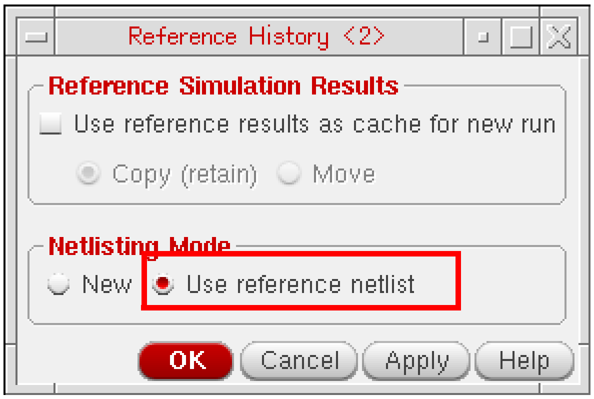
   Note: You can also replace the existing netlist file with another netlist for the same design.

5. In the ADE XL environment, click *Reference History Options* on the Reference History toolbar.

   The Reference History form is displayed.

6. In the Reference History form, select the *Use reference netlist* netlisting mode, as shown below.



When the *Use reference netlist* mode is set, ADE XL does not generate netlist during the next simulation run. Instead, it uses the netlist file available in the results directory of the referenced history name. In this case, this is the netlist file that was edited in **step 3**.

7. Click *OK* to close the form.

8. From the *Reference* drop-down list on the Reference History toolbar, select the name of a reference history corresponding to the edited netlist file.

9. Run simulation.

ADE XL will copy the edited netlist file from the results directory for `Interactive.3` to the results directory for the new simulation run.

Submitting a Point

You can sweep variables and parameters by specifying values for them without actually modifying their values in the active setup and then submit the resulting points for evaluation. Submitting points for evaluation allows you to experiment with different values for global variables and parameters without modifying the active setup.

You can submit points using any of the following two methods:

- Using the Single Run, Sweeps and Corners options form

  In this method, you submit a point based on the values of variables and parameters in the active setup that are modified as required. For more information, see Submitting a Point Using the Single Run, Sweeps and Corners Options Form on page 505.

- Submit points from a history item

  In this method, you submit a point based on the values of variables and parameters for a design point in a history item that are modified as required. For more information, see Submitting a Point From a History Item on page 509.

For more information, see the following topics:

Submitting a Point Using the Single Run, Sweeps and Corners Options Form

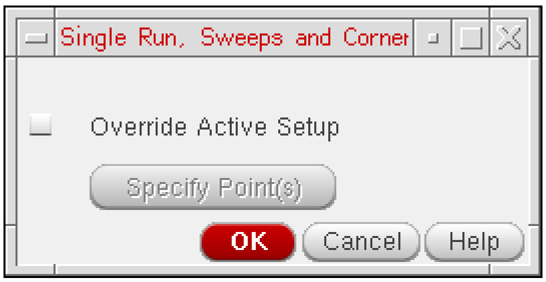
To submit a point using the Single Run, Sweeps and Corners options, do the following:

1. From the *Run* menu, select *Single Run, Sweeps and Corners*.

Tip

Alternatively, select *Single Run, Sweeps and Corners* in the *Select a Run Mode* drop-down list on the Run toolbar, then click the *Simulation Options*  button on the .bar.

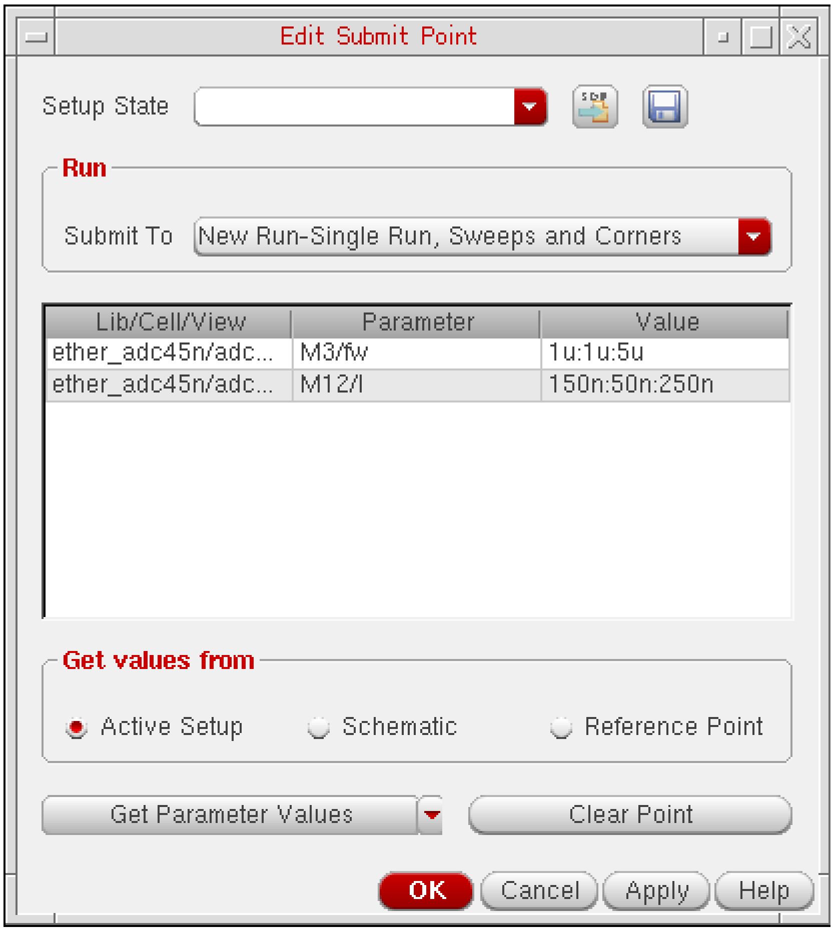
The Single Run, Sweeps and Corners form appears.



2. Select the *Override Active Setup* check box.

3. Click the *Specify Point(s)* button.

The Edit Submit Point form appears. If you have not previously created a submit point, this form will be populated with the active setup values.



The *Submit To* field in the *Run* section is by default set to `New Run-Single Run, Sweeps and Corners`. You can also submit a new point to an existing manual tuning run. For more details, refer to Manual Tuning in the *Analog Design Environment XL user guide*.

4. Add or modify the values as desired. You can specify your own values for the global variables and parameters, or load values from the active setup, schematic or an existing reference point.

- ❑ Select the *Value* field for any global variable or parameter and enter a value.

- To load values from the active setup, schematic or an existing reference point for global variables and parameters, do one of the following:

Select	To
Active Setup	Load values of global variables and parameters from the active setup.
Schematic	Load values of parameters from the schematic.
Reference Point	Load values of parameters from an existing reference point. **Note:** This option will be disabled if no values exist for the parameters in the reference point.

  Then, do the following to load the values:

  - To load the values for specific global variables and parameters, select the global variables and parameters for which you want to load the values, click *Get Parameter Values* and choose *Selected Parameters*.

    You can also select global variables and parameters, right-click and choose *Get Schematic Value*, *Get Ref Point Value* or *Get Active Setup Value* to load values from the active setup, schematic or an existing reference point for the selected global variables and parameters.

  - To load the values for all the global variables and parameters that have no values, click *Get Parameter Values* and choose *Parameters With No Value*.

  - To load the values for all global variables and parameters, click *Get Parameter Values* and choose *All Parameters*.

- Select one or more rows and click *Clear Point* to clear the values.

  You can also select rows, right-click and choose *Clear* to clear the values.

5. Click *OK* to save the changes and close the Edit Submit Point form.

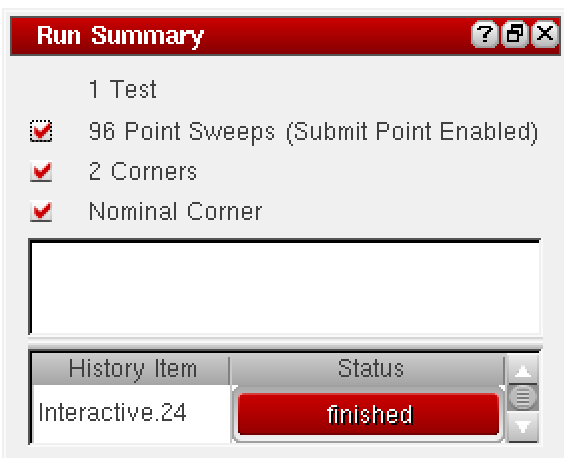
The submit point will remain the active submit point for the current session until you modify it. For more information, see Viewing and Modifying the Current Point in the Submit Point Form on page 511.

Important

The submit point information will not be automatically saved in the active setup and hence will not be available when you restart ADE XL. You must manually save the submit point information (see Saving a Point in the Submit Point Form on page 513) if you want to load it later (see Loading a Point in the Submit Point Form on page 515).

6. Click *OK* to close the Single Run, Sweeps and Corners form.

   The text *Submit Point Enabled* in the Run Summary assistant pane indicates that the submit point is enabled.



7. On the Run toolbar, click the *Run Simulation*  button to submit the sweep for evaluation.

The submitted point is simulated across all the enabled tests and corners and the results are displayed in the Results tab of the Outputs pane. Information about the submitted point is stored in a new history item.

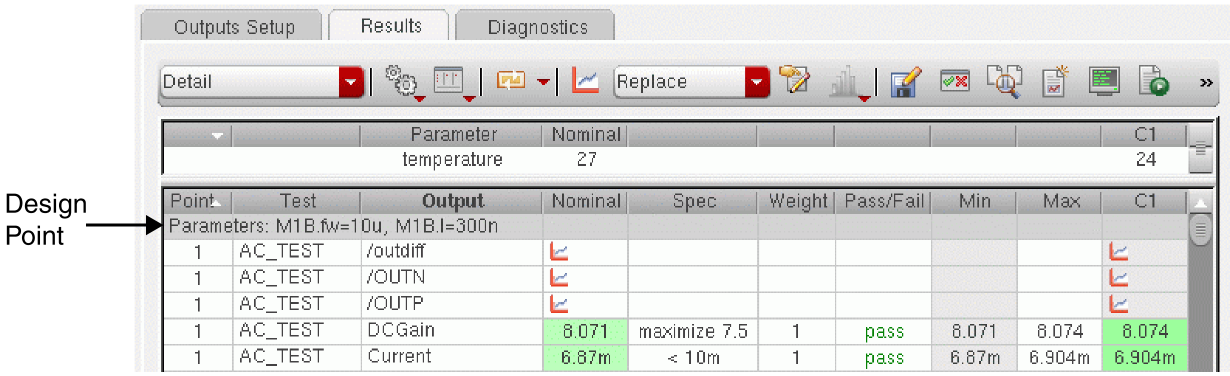
Submitting a Point From a History Item

To submit a point from a history item, do the following:

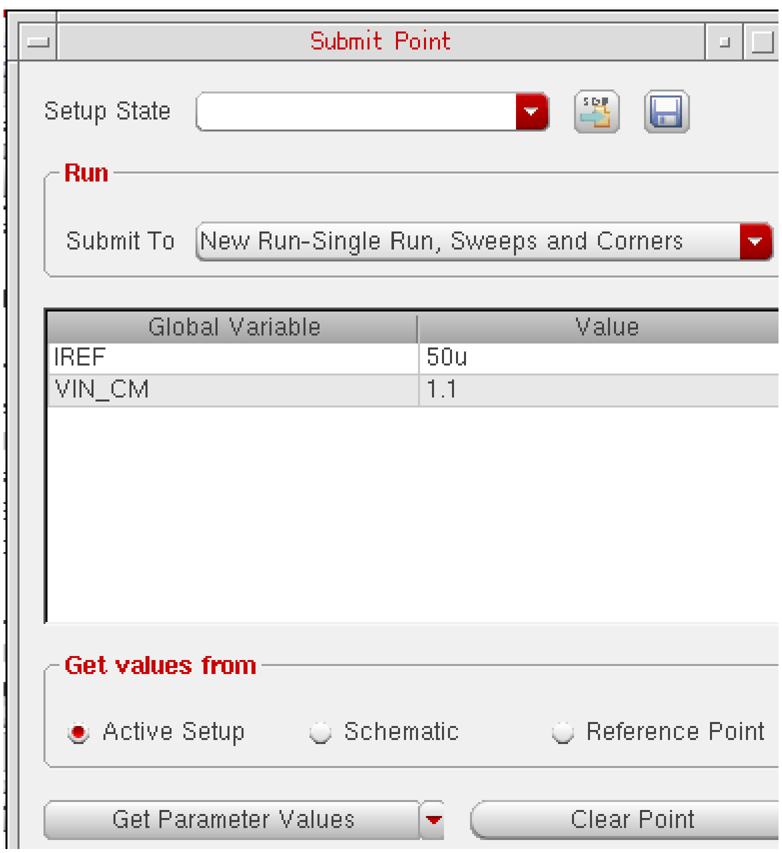
1. Display the results for the history item in the Results tab of the Outputs pane. For more information, see Viewing Results from a Particular Checkpoint on page 816.

2. Right-click on a design point (highlighted in gray) in the Results tab of the Outputs pane and choose *Submit Point*.

Note: You cannot submit a design point that contains statistical parameters. Hence, the *Submit Point* shortcut menu will be disabled if your design point contains statistical parameters (for example, statistical parameters from Sensitivity Analysis runs).



The Submit Point form appears displaying the values of global variables and parameters at the design point.



Note the following:

- ❑ Global variables or parameters that exist in the history item but not in the active setup will be highlighted in yellow color and automatically removed from the submit point when you click *OK* or *Apply* in the Submit Point form.
- ❑ Global variables or parameters that exist in the active setup but not in the history item are automatically added in the Submit Point form. You must specify values for such variables and parameters.

3. If desired, modify the values as described in step 4 of Submitting a Point.
4. Click *OK* to submit the point for evaluation.

The submitted point is simulated across all the enabled tests and corners and the results are displayed in the Results tab of the Outputs pane. Information about the submitted point is stored in a new history item.

Viewing and Modifying the Current Point in the Submit Point Form

The current point is the last one created or modified during the current session. You can view and modify this point.

To view and modify the current point, do the following:

1. From the *Run* menu, select *Single Run, Sweeps and Corners*.

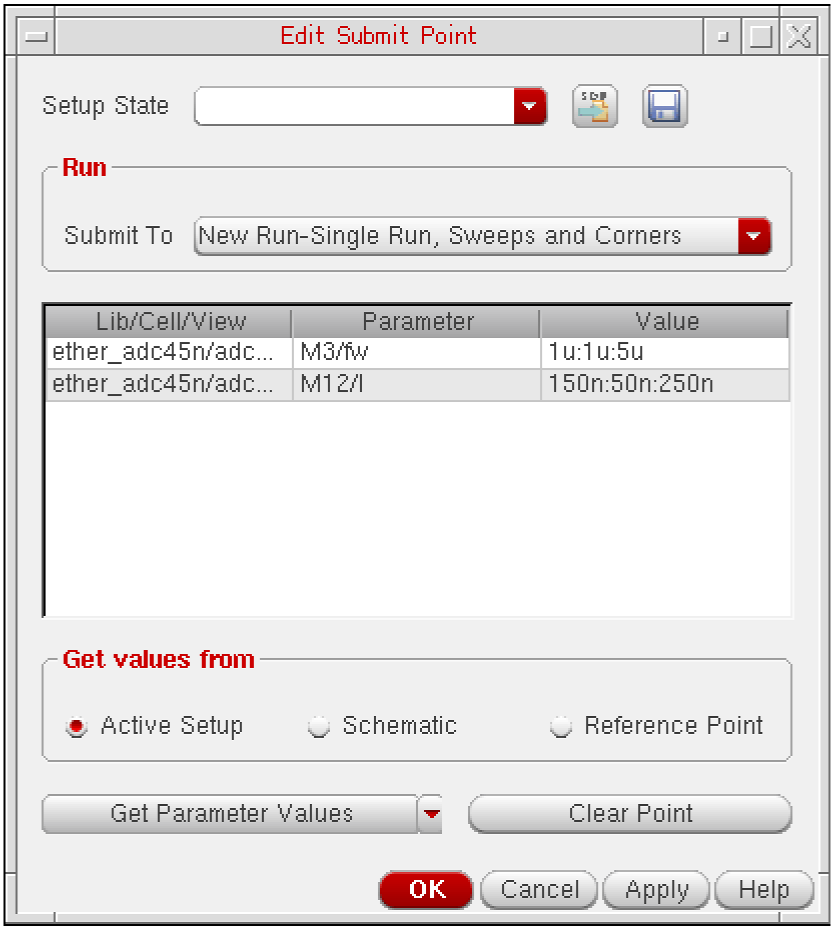
Tip

Alternatively, select *Single Run, Sweeps and Corners* in the *Select a Run Mode* drop-down list on the Run toolbar, then click the *Simulation Options*  button on the Run toolbar.

The Single Run, Sweeps and Corners form appears.

2. Click the *Specify Point(s)* button.

The Edit Submit Point form appears displaying the values for the current point.



Important

Global variables or parameters that exist in the point but not in the active setup are highlighted in yellow color and you will be prompted to enable or add them in the design space. If you click the *OK* button without enabling or adding them in the design space, they will be automatically removed from the point.

3. If desired, modify the values as described in step 4 of Submitting a Point.
4. Click *OK* to save the changes and close the Edit Submit Point form.

Important

The point information will not be automatically saved in the active setup and hence will not be available when you restart ADE XL. You must manually save the point information (see Saving a Point in the Submit Point Form on page 513) if you want to load it later (see Loading a Point in the Submit Point Form on page 515).

5. Click *OK* to close the Single Run, Sweeps and Corners form.

Saving a Point in the Submit Point Form

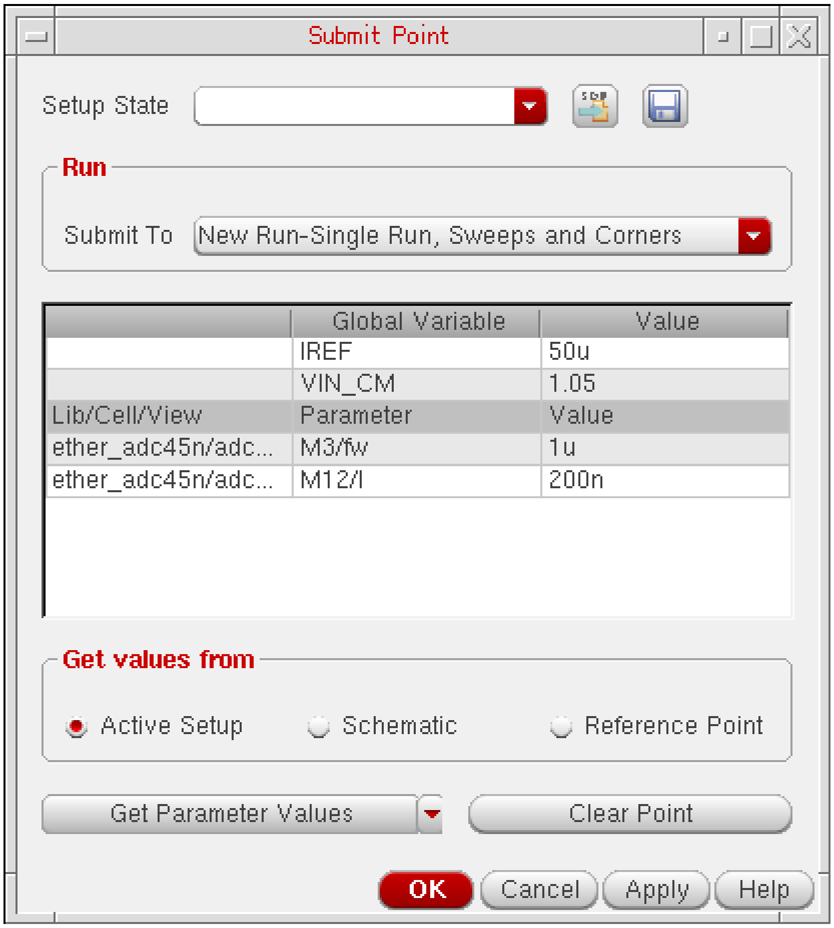
By default, the point information in the Submit Point form is not saved automatically in the active setup. Therefore, the point information in the current ADE XL session is not available when you restart ADE XL. You can save the point information to a setup state if you want to load it in a new ADE XL session.

Tip

You can also save different combinations of variable and parameter values to setup states and load them later.

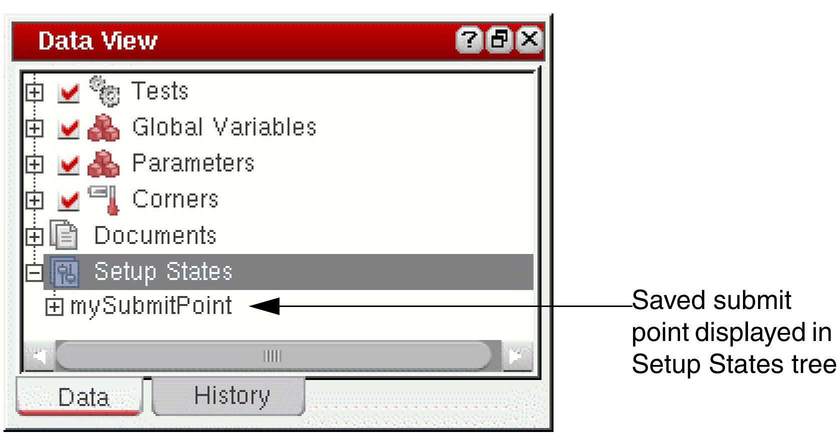
To save a point, do the following in the Edit Submit Point form:

1. In the *Setup State* field, type a name for the point.



2. Click .

The saved point is displayed in the *Setup States* tree on the Data View pane.



Loading a Point in the Submit Point Form

You can load a point in the Edit Submit Point form or in the *Setup States* tree in the Data View pane.

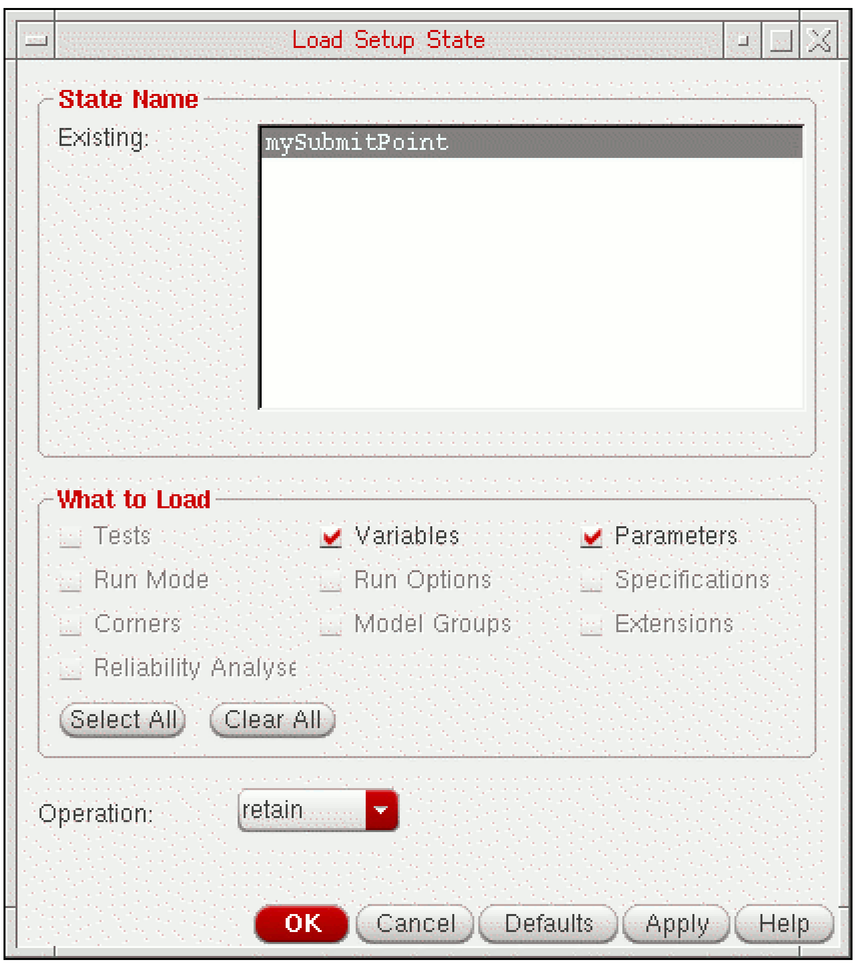
To load a point in the Edit Submit Point form, do the following:

➡ Select the point from the *Setup State* drop-down list and click .

To load a point in the *Setup States* tree in the Data View pane, do the following:

1. Right-click the point and choose *Load*.

   The Load Setup State form appears.



2. Select the point you want to load and click *OK*.

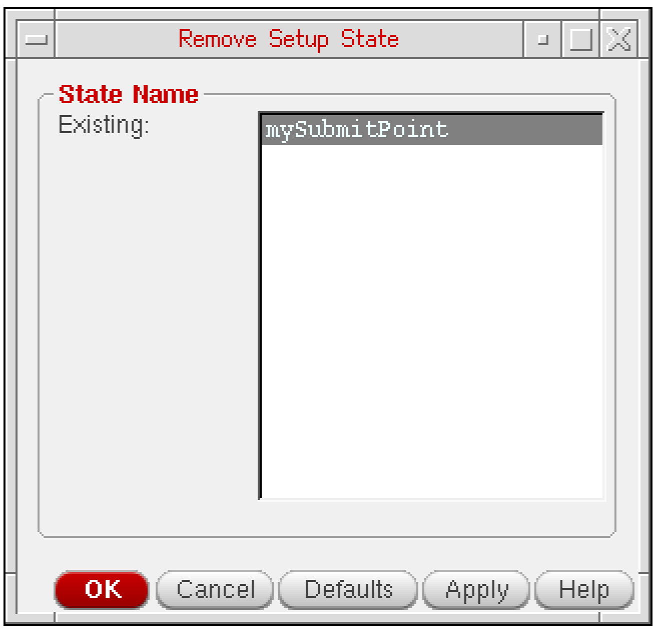
For more information about the Load Setup State form, see Loading a Setup State on page 787.

Deleting a Point

To delete a saved point, do the following:

1. In the *Setup States* tree in the Data View pane, right-click the point and choose *Delete.*

   The Remove Setup State form appears.



2. Select the point you want to delete and click *OK*.

   For more information about the Remove Setup State form, see Deleting a Setup State on page 790.

Simulating Only Error or Incomplete Points

If a history item includes points that are incomplete or that show errors, it could be because the simulation run was stopped before completion, or some simulations failed due to issues like license check failures, an unavailable model file, system issues on a remote host used for distributed simulation etc. For such history items, you can run incremental simulations to simulate only the incomplete or error points in the history item. This helps in saving time because the results for the completed points will be reused from the previous history item and the simulations will not run again for those points.

ADE XL provides two ways in which you can run only those points that had errors or that were incomplete in the previous run:

- By using the Re-run Unfinished/Error Points Command
- By setting reference to the simulation results of a previous run

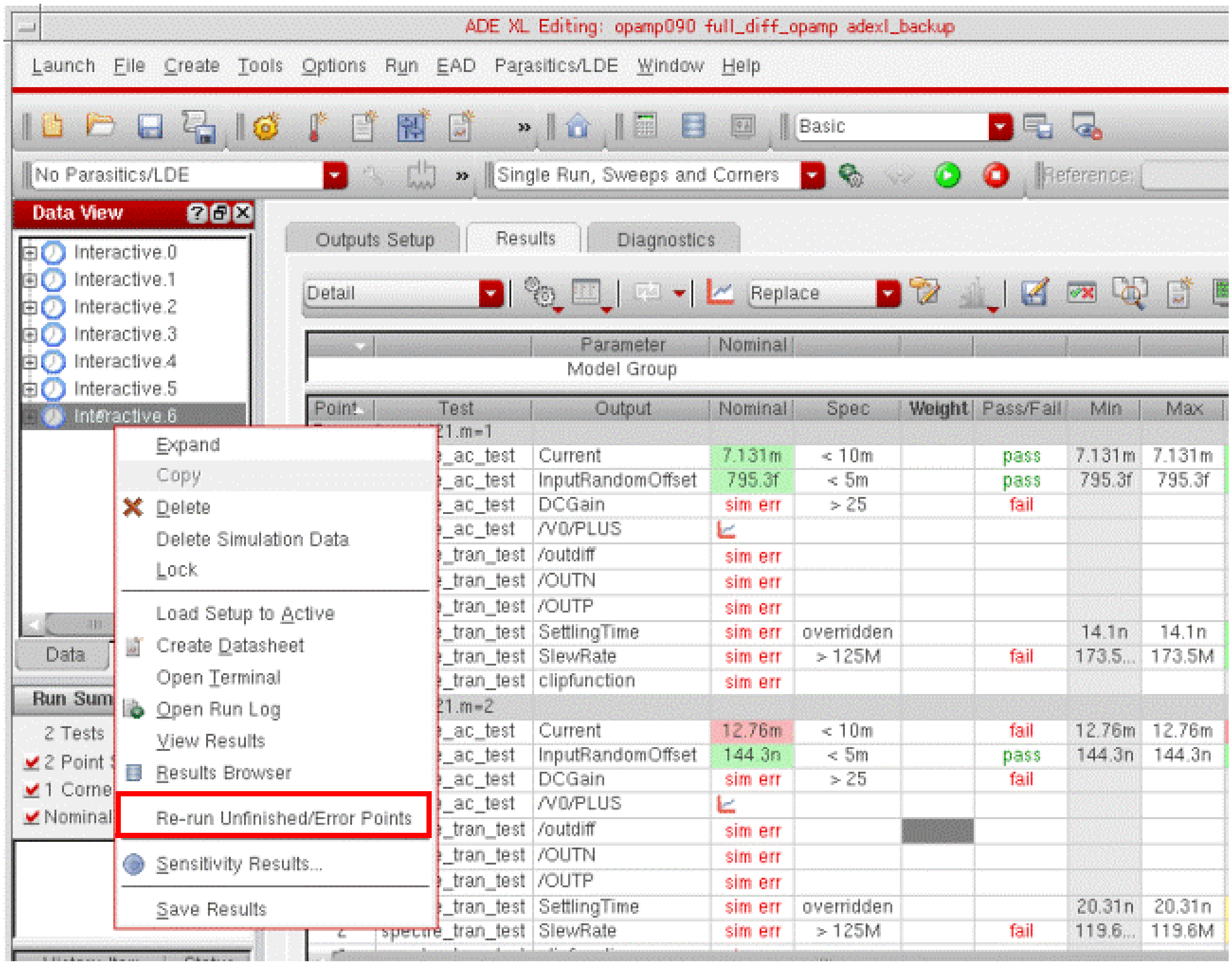
Method 1: Running Error or Incomplete Points By Using the *Re-run Unfinished/Error Points* Command

Note: This method is useful in scenarios when the design has not changed and you want to use the previous simulation setup while completing the unfinished points. If the design has changed, a new netlist will be generated, which may be different from the one used in the original run with unfinished points.

To simulate only the incomplete or error points in a history item, do the following:

1. Ensure that the History tab of the `Data View` assistant pane is open.

2. Right-click on the desired history item that contains error points and choose *Re-run Unfinished/Error Points*.



When you use this command, ADE XL ignores any changes in the active setup after the last run and re-runs simulations for all the error or unfinished points in the selected history run. All the consolidated results are saved in a new history named `history_name.rerun.seqNum`, where `seqNum` starts from `0` and increases incrementally in the subsequent simulation runs.

Note the following:

- The simulation run uses the setup of the history item on which you ran *Re-run Unfinished/Error Points.* So any changes you make in the active setup will be ignored during the run.

- When you run *Re-run Unfinished/Error Points* on a history item, the simulation results of that history item are moved to the new history item. As a result, you will not be able to perform postprocessing operations (like plotting, printing, annotation, re-evaluation, and so on) on the history item on which you ran *Re-run Unfinished/Error Points*. You can use the `retainReferenceSimResults` environment variable to retain the simulation results of the history item on which you ran *Re-run Unfinished/Error Points*. For more information about this environment variable, see retainReferenceSimResults.
- You cannot run *Re-run Unfinished/Error Points* on child history items. For more information about child history items, see Viewing Checkpoints.

Method 2: Running Error or Incomplete Points By Referencing the Simulation Results of a Previous Run

Note: This method is useful when the design has changed, but you want to reuse the previous netlist and simulation setup while completing the unfinished points.

In this method, you can reference the netlist and the existing simulation results of a previous run. For details on the pre-requisites of this method and steps to run an incremental simulation, refer to Running an Incremental Simulation.

Troubleshooting a Design or Data Point

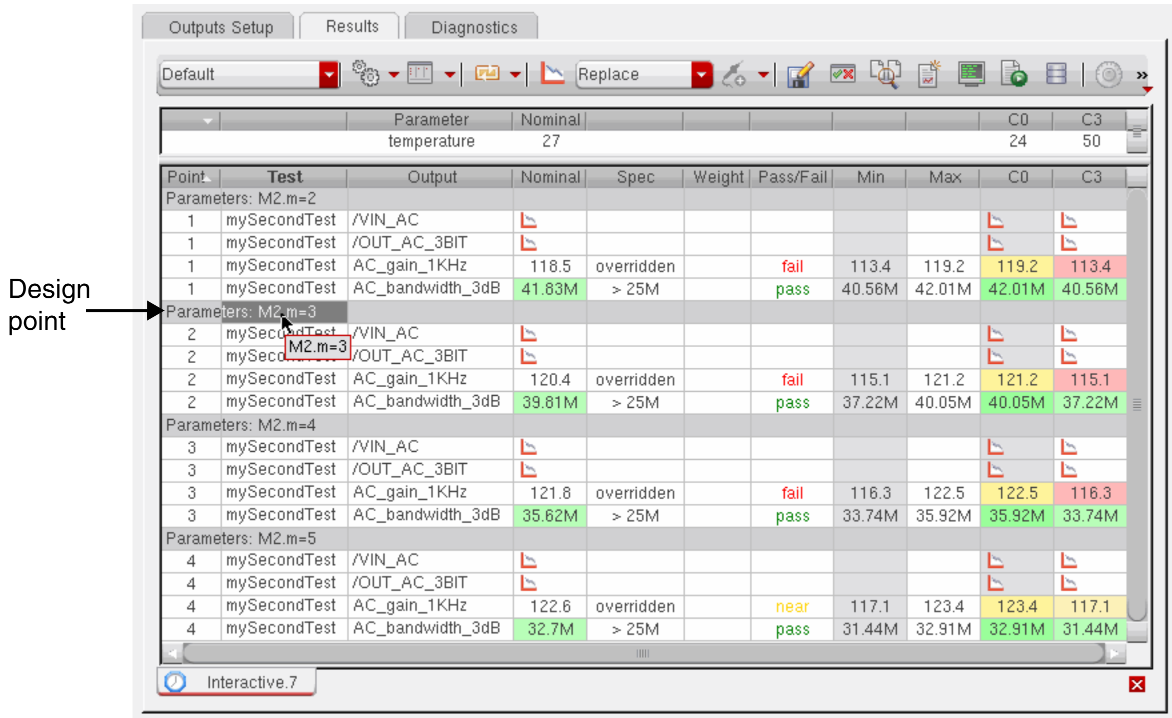
When you troubleshoot a design point or data point, ADE XL submits the point for evaluation, then saves out the files necessary for you to produce results in your simulator. These files are specific to each simulator, but generally include the following that can be used to troubleshoot problems in points:

- your current environment settings
- logs of commands run by the simulation nodes
- a log of interactions with the simulator
- the changelist sent to the simulator
- the netlist that ADE XL will simulate
- the simulation results

To troubleshoot points, do one of the following on the Results tab of the Outputs pane:

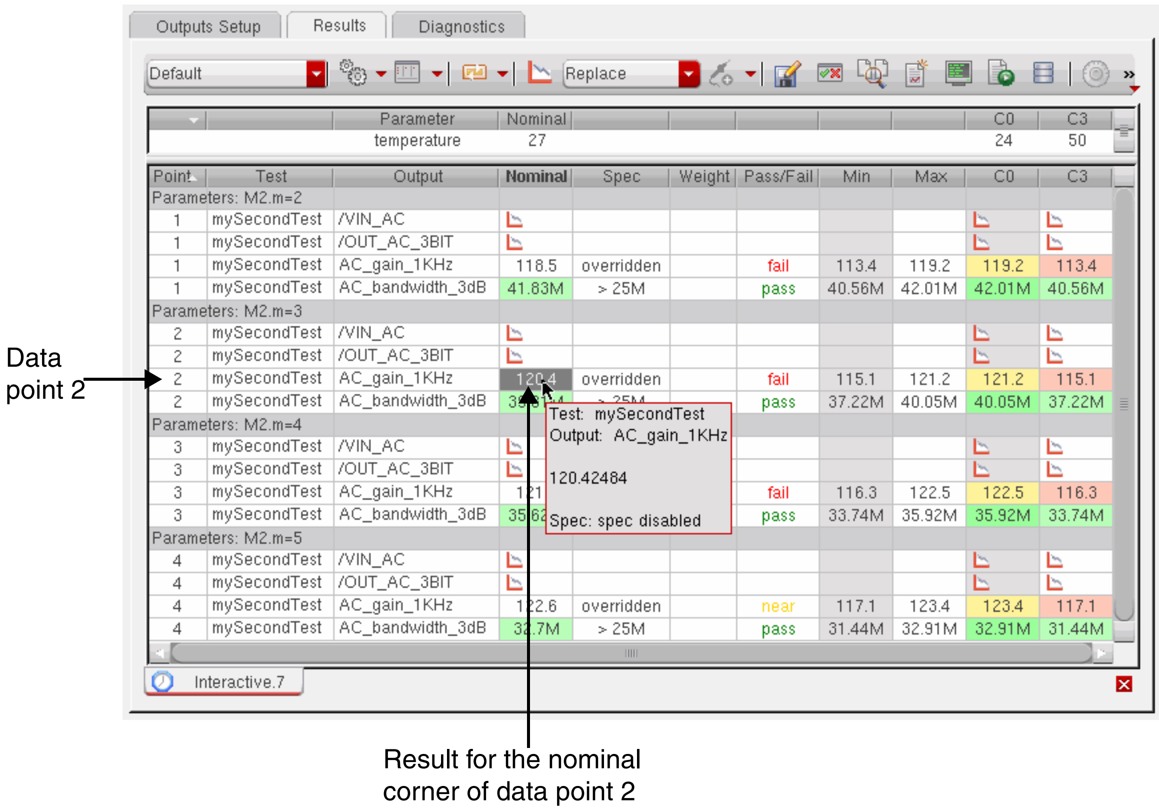
- To troubleshoot an entire design point (across all corners), right-click on the design point and choose *Troubleshoot Point*.

The following figure shows an example of design point `M2.m=3` being selected.



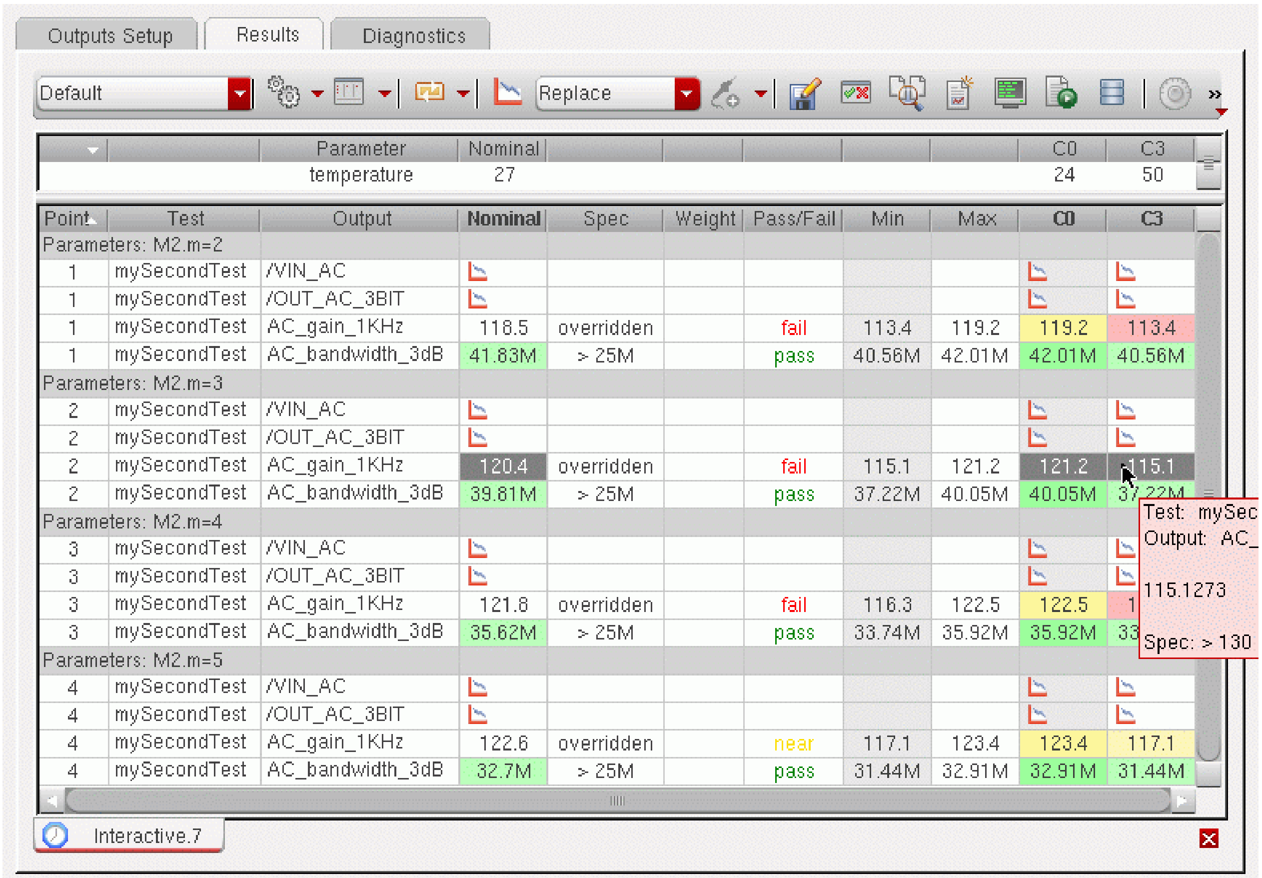
- To troubleshoot a data point at a corner, right-click on the result for the data point at the corner and choose *Troubleshoot Point*.

The following figure shows an example of the result for the nominal corner of data point `2` being selected.



■ To troubleshoot a data point at more than one corner, hold down the *Ctrl* key, click on the results for the data point at the corners, then right-click and choose *Troubleshoot Point*.

The following figure shows an example of the results for the nominal corner and corners `C0` and `C3` of data point `2` being selected.



Tip

To troubleshoot a point from the results of a history item, display the results for the history item in the Results tab of the Outputs pane (see Viewing Results from a Particular Checkpoint on page 816), right-click on the point in the Results tab and choose *Troubleshoot Point*.

ADE XL submits the point for evaluation, then saves all the information for that point to a history item named `history_name.TS.seqNum`, where `seqNum` is `0` (zero) for the first item, then 1+(the largest existing `seqNum` for that `history_name`). For example, if you run troubleshoot point on a history item named `interactive.9`, the new history item will be named `interactive.9.TS.0`.

The results directory for the new history item will have subdirectories for each corner that contain the following data for troubleshooting the point:

Directory Name	Contents
netlist	Netlist and related files.
psf	Simulation results.
troubleshoot	■ `Job.log` file. For more information about this file, see Viewing the Job Log on page 489. ■ `changelist` file that contains the list of changes in values of global variables, parameters and corners that are passed to the simulator. ■ `environ` file that contains the current environment settings on the remote node. ■ `evalshell` file that can be used to start a shell using the `environ` file. ■ `ocean.il` file that contains the OCEAN commands that can be used to troubleshoot expression evaluation errors.

Note: When you run troubleshoot points, ADE XL ignores the settings in the Save Settings form (see Specifying Options for Saving Simulation Results on page 73).

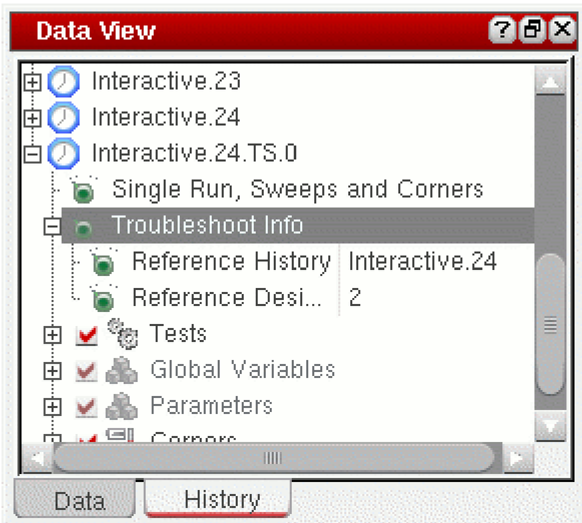
To view information regarding the design or data point for which you ran troubleshoot point, do the following:

1. On the History tab of the `Data View` assistant pane, click + to expand the history item for the troubleshoot point.

   The history details appear in the expanded branch.

2. Click + next to *Troubleshoot Info* branch.

   The information regarding the design or data point for which you ran troubleshoot point is displayed. For example, the *Troubleshoot Info* branch in the following figure indicates that you ran troubleshoot point on the second design point of the history item named `Interactive.24`.



Debugging Points

After you run simulations in ADE XL, if any particular data point fails or shows undesired results, you can selectively change the setup of that data point and debug it. For this, you can use the ADE XL Debug Environment.

The ADE XL Debug Environment is a Virtuoso Analog Design Environment L (ADE L) based debug environment where you can load the setup of a data point you want to debug. The entire setup for the data point selected in ADE XL including the test, variables, and analyses details, corners, outputs, simulation settings, and job policy gets loaded. You can modify the settings for debugging purpose and perform all the tasks that you can perform in ADE L, such as running simulations and plotting results, except the following tasks:

- Saving and loading states
- Changing the simulator or simulation run mode
- Changing the switch view list
- Running parametric analysis

Note: Simulations that run from the debug environment do not use ICRP. They are same as ADE L simulations.

After debugging the data point, you can bring back the corrected simulation setup to the ADE XL view. You can then run the simulation by using the updated setup in the ADE XL view.

Video

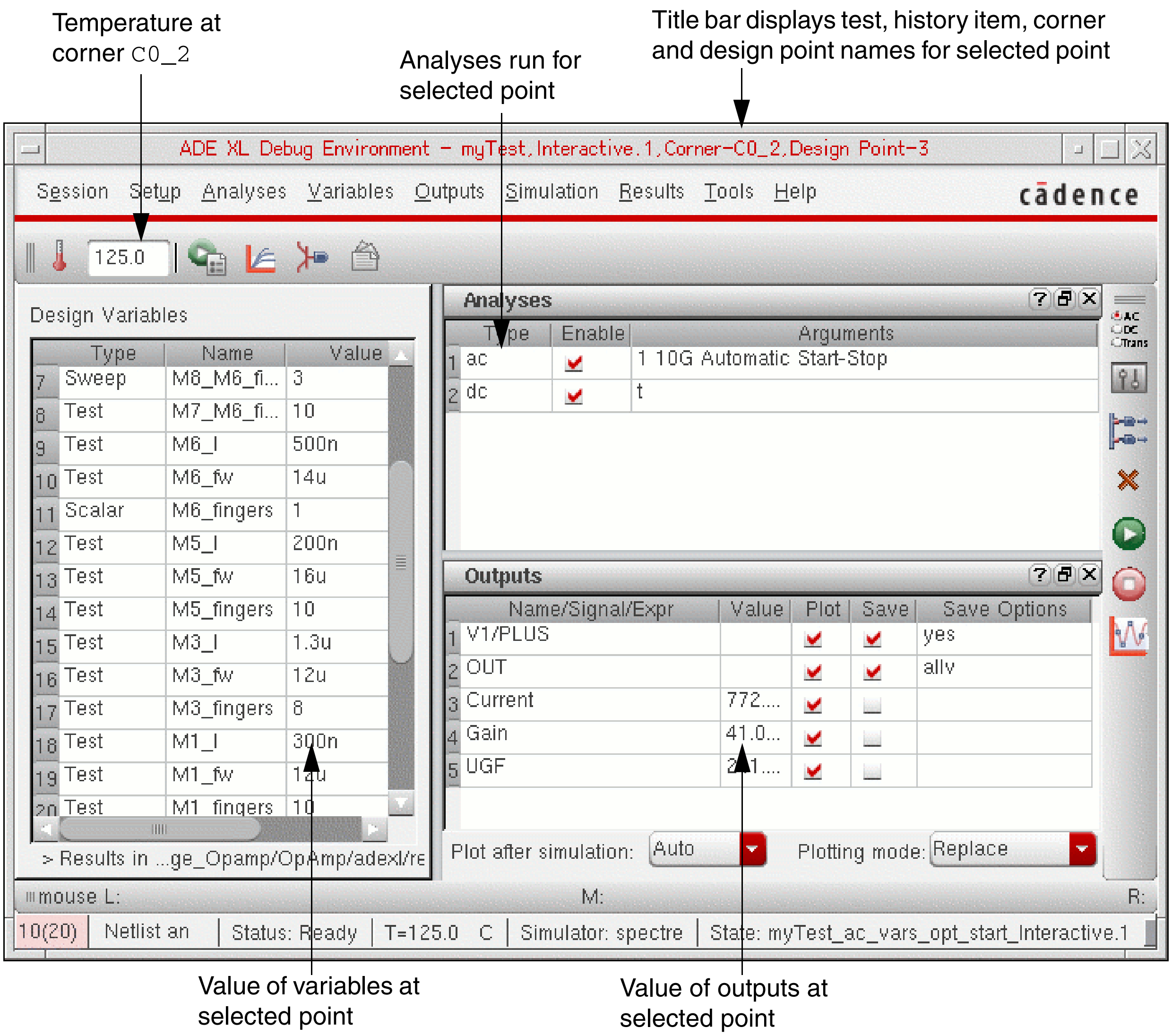
You can view video demonstration for this feature at Debugging Points of Simulation Results in ADE XL.

To debug a data point by using the debug environment, do the following:

1. On the Results tab, right-click the data point and choose *Open Debug Environment*.

   The ADE XL Debug Environment window appears displaying the setup for the data point. For example, the following figure displays the setup for corner `C0_2` of test `myTest` at data point `3` in history item `Interactive.1`.

Figure 13-3 ADE XL Debug Environment



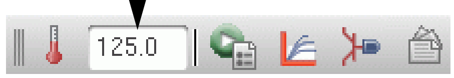
Note the following:

- The ADE XL Debug Environment window title bar displays the test name, history item name, corner name and design point ID for the selected point. For example, the title bar in Figure 13-3 on page 526 displays the point information as:

  `myTest, Interactive.1, Corner-C0_2, Design Point-3`

  Where, `myTest` is the test name, `Interactive.1` is the history item name, `C0_2` is the corner name and `3` is the design point number for the selected point.

- The *Design Variables* pane displays the value of the design variables at the selected data point. The *Type* column indicates the type of the variables, as described below:

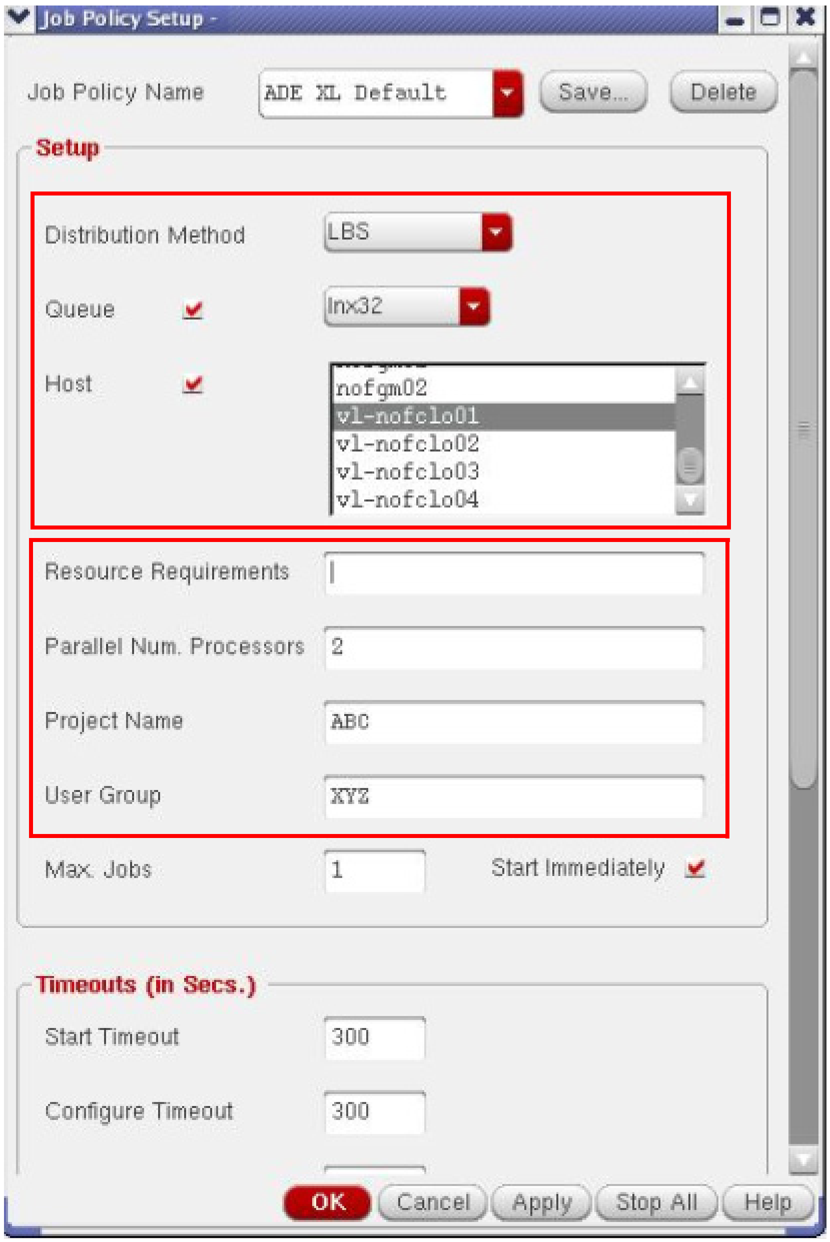
Type	Description
Test	Design variable from test
Sweep	Global variable or device parameter with sweep (multiple) values
Scalar	Global variable or device parameter with single value
Constraint	Matched or ratioed device parameter **Note:** You can modify only the value of the master parameter. Values of the matched and ratioed device parameters will change accordingly.
Corner	Design variables and parameters specified for the corner **Note:** The value of the *Temperature* variable specified for the corner is displayed on the toolbar, as shown below: Temperature at corner `C0_2` 

- The default project directory for the data point is the `/results/data/`*`historyName`*`/`*`pointID`*`/`*`testName`*`/debug` directory of the ADE XL view.

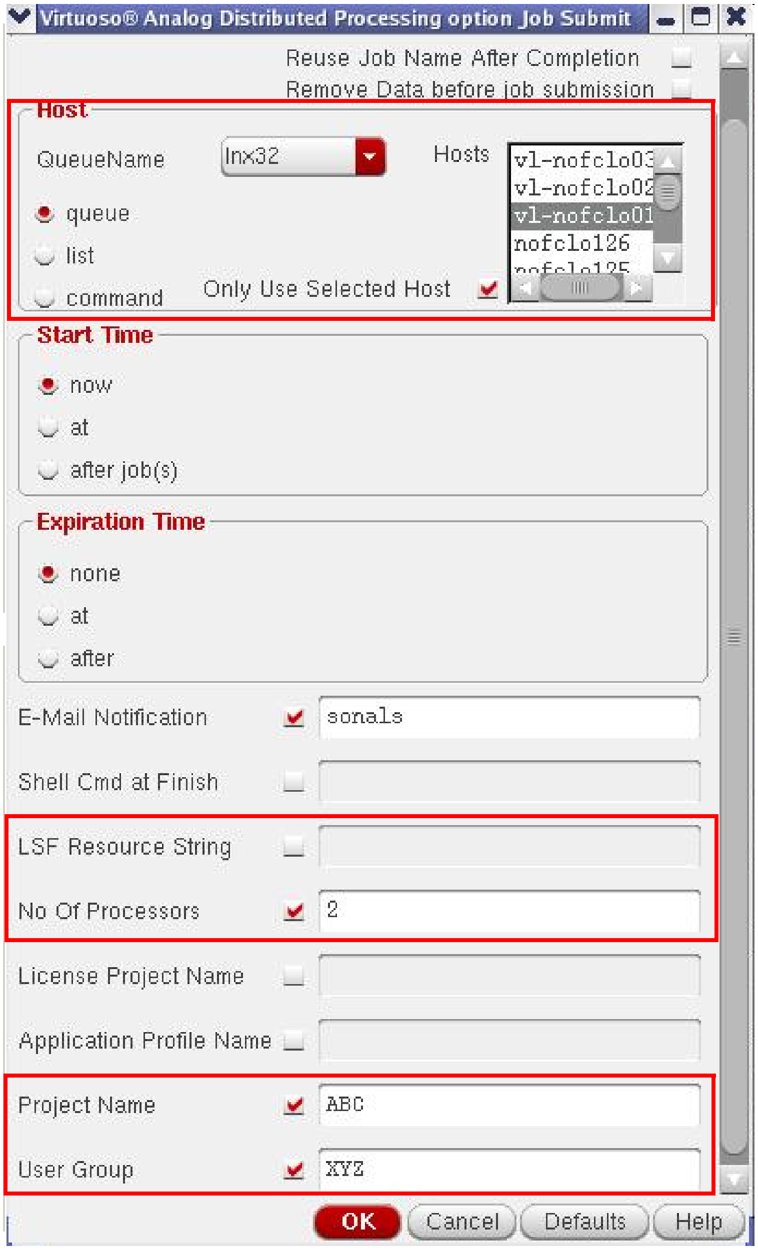
  Note: You can choose *Setup – Simulator/Directory/Host* to modify the project directory path.

- You can see the path of the results directory for the data point in the *ADE Result Directory* toolbar. By default, this toolbar is hidden. You can right-click on any toolbar and choose *ADE Result Directory* to make it visible.

- The relevant job policy settings set up in the ADE XL environment are copied to the debug environment. For example, if job policy is set in the ADE XL environment as shown below:



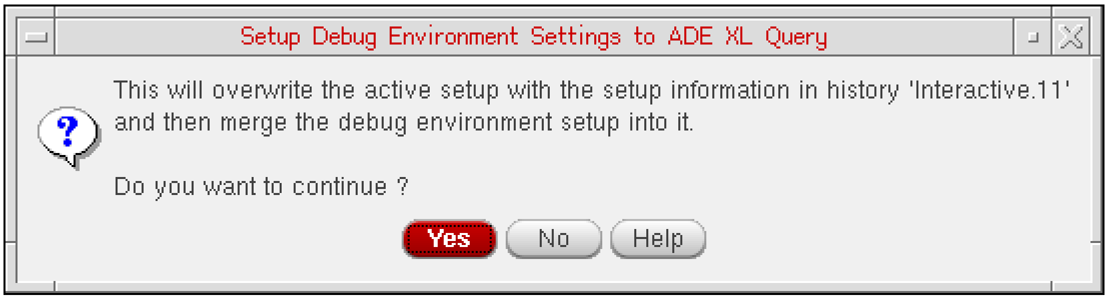
Note that in the form shown above, settings are done for distributed processing by using the LBS method. Relevant settings for distributed processing are copied from this form to the Analog Distributed Processing option Job Submit form in the debug environment, as shown below:



- You cannot use the debug environment to debug results from OCEAN-based tests. For more information about OCEAN-based tests, see Working with OCEAN-Based Tests on page 125.
- Any existing `.cdsinit` variables set by the user (for example, the variables to set the simulation directory or the host run mode) are applicable to the debug environment.

2. Modify the setup for the data point as required and run simulations to debug the data point.

3. (Optional) Choose *Session – Setup Back To ADEXL* and click *Yes* in the message box that appears to bring the corrected setup back to the ADE XL view.



If you click *Yes*, ADE XL overwrites the active setup with the setup information in the history item for which you debugged the data point. For example, if you debug a data point in a history item named `Interactive.2`, ADE XL overwrites the active setup with the setup information in `Interactive.2`. Then, the changes in the debug environment setup are merged with the active setup as described below:

- ❑ Changes in design variables, analyses and outputs for a test in the debug environment are merged with the setup for that test in the ADE XL view.

- ❑ Changes in scalar variables (global variables or device parameters with single value) in the debug environment overwrite the value of the corresponding global variables and device parameters in the ADE XL view.

- ❑ Changes in sweep variables (global variables or device parameters with sweep values) in the debug environment are appended to the sweep range for the corresponding global variables and design parameters in the ADE XL view.

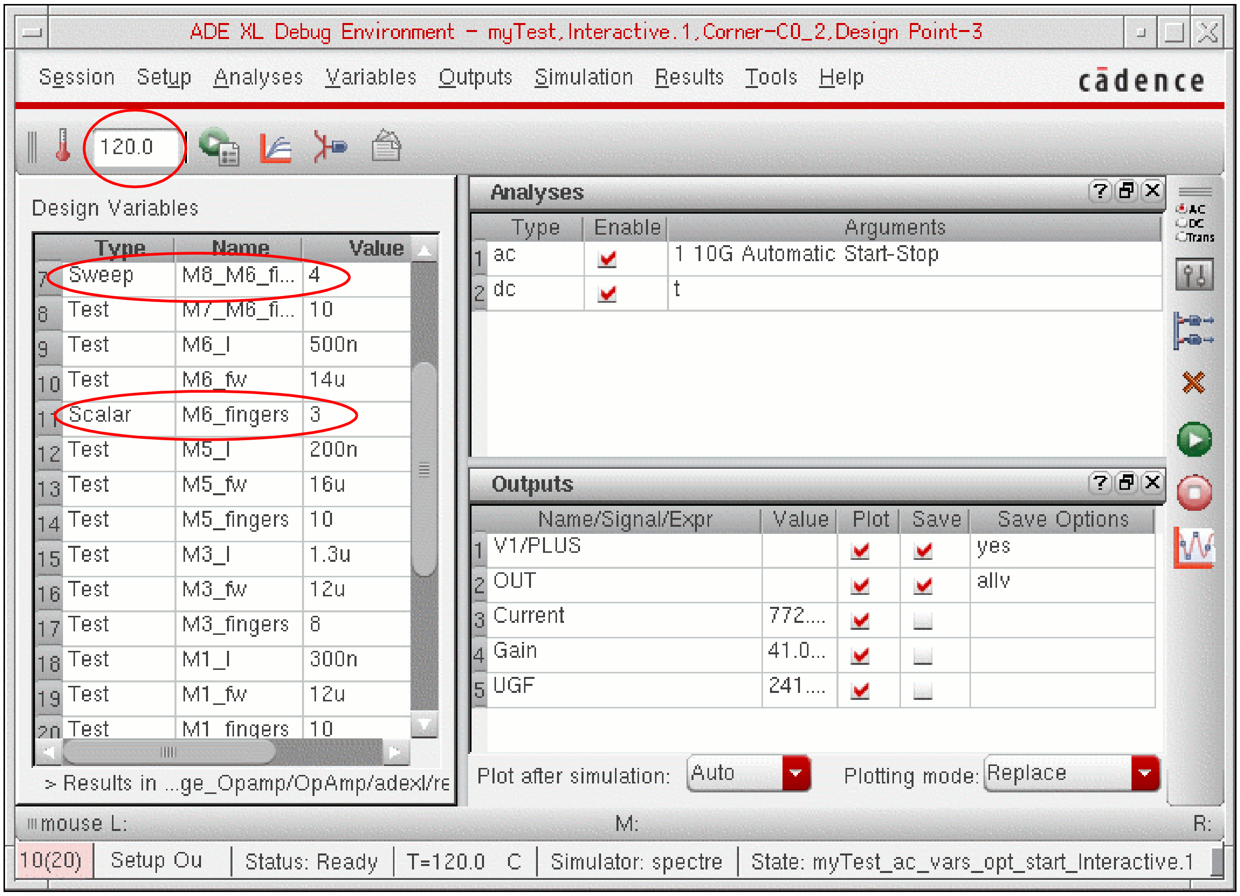
- ❑ Changes in corner values in the debug environment result in a new corner being created in the ADE XL view. The new corner is enabled only for the test for which you debugged the data point.

For example, if you choose *Session – Setup Back To ADEXL* after making the following changes in the debug environment, as shown in the figure:

- ❑ Modified the value of the scalar variable `M6_fingers` from `1` to `3`

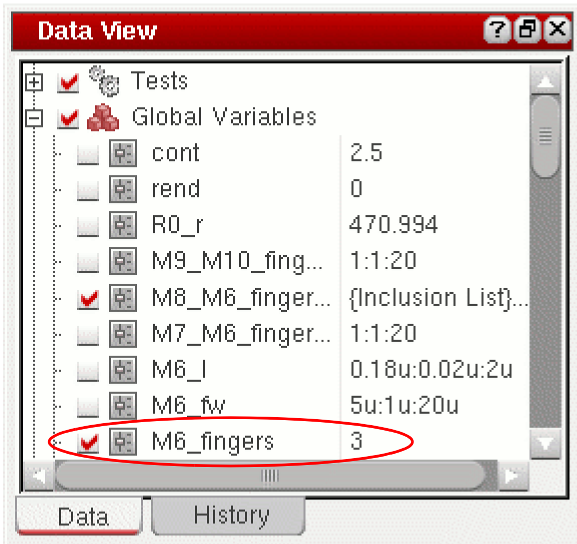
- ❑ Modified the value of the sweep variable `M8_M6_fingers_ratio` from `3` to `4`

❑ Modified the value of the *Temperature* variable for the corner from `125` to `120`

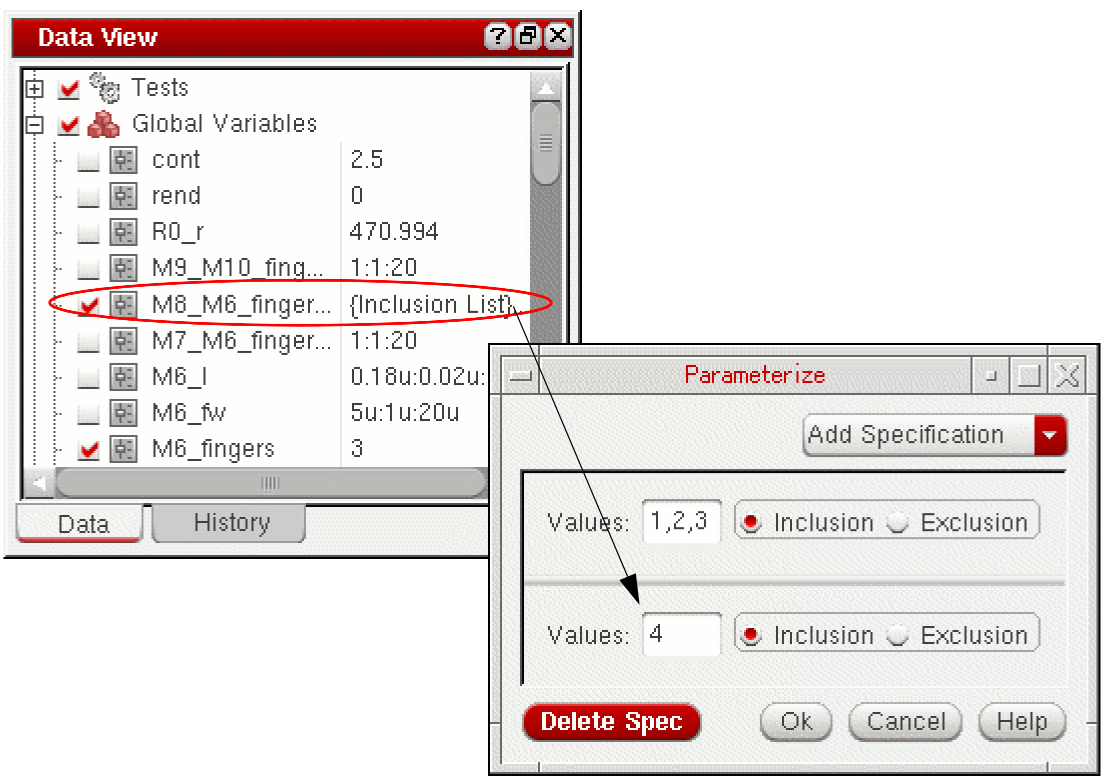


The changes are merged into the active setup as described below:

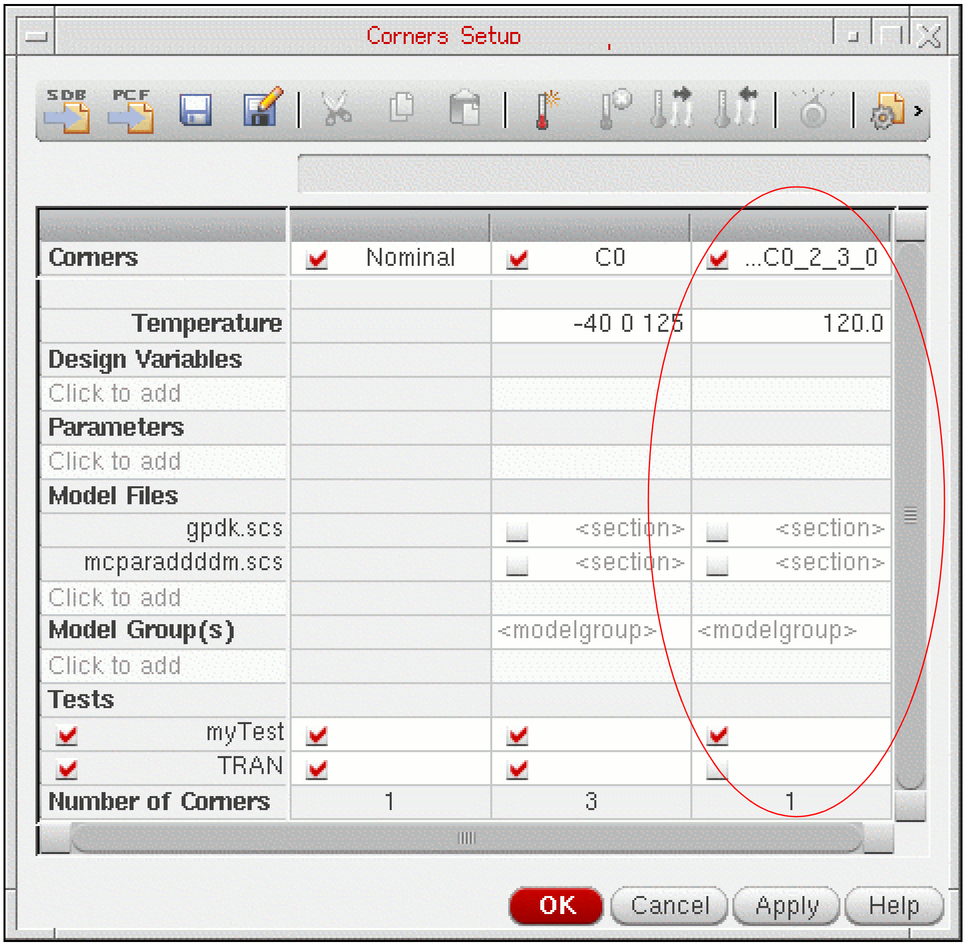
- ❑ The value of the scalar global variable `M6_fingers` changes to `3`.



- ❑ The value `4` specified for the sweep global variable `M8_M6_fingers_ratio` is included in the sweep range for the variable.



- ❑ A new corner named `C0_2_3_0` which is enabled for test `myTest` for which you debugged the data point is created because of the change in the value of the *Temperature* variable for corner `C0_2` from `125` to `120` in the debug environment.



4. (Optional) Run the simulation with the modified setup.

Important Point to Note

It is important to note the following point while working in the debug environment:

- If you try to open a debug environment for a point while this environment is already open for another point, the set up of the previous point is replaced with the new point. However, if a simulation is already running for the previous point, the following confirmation message is displayed.

Click *No* to continue with the simulation already running.

Click *Yes* to stop the simulation that is already running. The simulation running from the debug environment is stopped and the partial simulation results are plotted. In addition, the setup details of the new debug point overwrite the setup of the previous data point.

Creating and Running an OCEAN Script

You can create and run an OCEAN script to run one or more tests over zero or more corner conditions. When you save an OCEAN script, the program saves setup information for all your tests, sweeps, corner conditions and Monte Carlo analyses.

For more details, refer to the following sections:

- Creating an OCEAN Script
- Modifying an OCEAN Script
- Running an OCEAN Script
- Running Parallel OCEAN XL Simulation Runs for an ADE XL View
- Viewing Results of Simulations Run using OCEAN Scripts

Creating an OCEAN Script

You can create an OCEAN script in any of the following two ways:

- ➡ Saving the ADE XL Setup to an OCEAN Script
- ➡ Scripting an OCEAN Script
- ➡ Creating OCEAN Scripts with Netlist Specified as a Design

Saving the ADE XL Setup to an OCEAN Script

You can create a simulation setup in ADE XL user interface and save it as an OCEAN script. For this, after the simulation setup is complete, do the following:

1. In the main session window, choose *File – Save Script*.

   The Save OCEAN Script form appears.

2. In the *File Name* field, type a name and location for your OCEAN script file.

Note: The default name and location is `./oceanScript.ocn`.

3. Click *OK*.

   The program saves simulation setup and conditions to the specified OCEAN script file.

Note: The setup information for non-ADE XL tests will not be saved in the OCEAN script.

Scripting an OCEAN Script

You can use a text editor to create an OCEAN script by using OCEAN commands and save it in a `.ocn` file.

Creating OCEAN Scripts with Netlist Specified as a Design

If you have an already generated netlist of a design, you can provide the path to that netlist file to the `design` OCEAN command. By doing this, you can run a simulation without loading/requiring the complete design hierarchy. Later, you can view the simulation results in ADE XL user interface. For more details, refer to Viewing Results of Simulations Run using OCEAN Scripts.

Modifying an OCEAN Script

If you run an OCEAN script from an ADE XL interface, if required, you can load its setup in ADE XL user interface and modify it after the run. To modify the OCEAN setup, do the following:

1. In the History tab of the `Data View` assistant pane, right-click on the history item for the OCEAN run and choose *Load Setup to Active*.

Caution

Load Setup to Active will overwrite the current setup with the setup in the history item for the OCEAN run.

2. Modify the setup as required.

   You can modify the setup using the ADE XL user interface (such as adding or modifying global variables or corners using the Data View assistant pane), or use the OCEAN Test Editor form to modify the setup.

   To use the OCEAN Test Editor form, do the following:

a. Right-click on a test name on the *Tests* tree in the Data tab of Data View assistant pane and choose *Open Test Editor*.

The OCEAN Test Editor form appears.

The OCEAN Test Editor form is a text editor that displays the OCEAN commands saved in a script.

b. Change the simulation setup by modifying the OCEAN commands in the OCEAN Test Editor form.

c. Click *OK* to save the changes in the setup.

Important

When you modify the setup in the history item for the OCEAN run in ADE XL, the original OCEAN script that you used to run the simulation from a UNIX shell will not be modified. Choose *File – Save Script* if you want to save the changes in the setup to a new OCEAN script file.

Running an OCEAN Script

To run simulations using OCEAN scripts, perform any one of the following steps:

➡ In CIW, run the following command:

```
load("<name-of-OCEAN-script-file>")
```

➡ In ADE XL user interface, load the set up from history of a previous OCEAN run and click *Run Simulation* on the Run toolbar to start the simulation.

➡ In a UNIX shell, type the following commands:

```
ocean
load("<name-of-OCEAN-script-file>")
```

For information about running an OCEAN script from a UNIX shell, see the *OCEAN Reference*.

When you run a simulation using an OCEAN script, a history item named `Ocean.n` is created for the run. You can then start ADE XL, and use the ADE XL user interface to view the results for the OCEAN run.

Important

You must not load and run your OCEAN script in the Command Interpreter Window (CIW) while ADE XL is still running for the same cellview.

When you run the script, the program reports the following information:

- Sweep parameters and their values
- Number of tests, sweep points, and corners
- Points completed and job status information
- Results location to the output area of the CIW

  For example:

```
1/1 completed.
*Info*   The result of this OCEAN XL run are saved in "Interactive.3" in library
"rfExamples", cell "ne600", view "adexl".
```

The results location corresponds to the lib/cell/view specified in the `ocnxlTargetCellView` call, such as

```
ocnxlTargetCellView( "rfExamples" "ne600" "adexl")
```

See "OCEAN Commands in XL Mode" in the *OCEAN Reference* for information about OCEAN script commands for ADE XL.

Tip

You can run an OCEAN script using LSF, for example, by submitting the job remotely using the `bsub` command (for the LSF example) as follows:

```
bsub ocean -nograph OCEANscriptFileName
```

See also Specifying a Job Submit Command for additional information.

Also see: includeSimLogInJobLog, an environment that can be used to control the inclusion of the simulator output log in the OCEAN job log.

Running Parallel OCEAN XL Simulation Runs for an ADE XL View

In OCEAN XL, you can simultaneously run multiple simulation runs for an ADE XL view. You can do this by executing multiple ocnxlRun functions in parallel from the same OCEAN XL script. The results of all the runs are saved in the results database of that ADE XL view.

This section describes how to prepare setup and run parallel simulation runs in OCEAN XL.

Preparing Setup for Parallel Simulation Runs

You can enable the parallel run option for your OCEAN XL scripts in any of the following ways:

- To enable the parallel run, before saving an OCEAN XL script from an ADE XL view, select the *Parallel* option in the **Run Options** form.

  See also:

  - Saving State Information
  - Setting Up Run Options

- Use the ocnxlSetRunDistributeOptions command in the OCEAN XL script to set the parallel run option.

Running Parallel Simulation Runs

In your OCEAN script, ensure that the *waitUntilDone* argument of the ocnxlRun function is set to `nil`.

For example, if you have saved an ADE XL state, `Ac_State1`, and want to run a simulation for it parallel to other runs in your OCEAN XL script, use the following commands:

```
ocnxlLoadSetupState( "AC_State1" 'retain ?tests t ?vars t ?parameters t
?currentMode t ?runOptions t ?specs t ?corners t ?extensions t
    ?modelGroups nil ?relxanalysis nil )

runid1 = ocnxlRun(?waitUntilDone nil)
ocnxlLoadSetupState( "Tran_State2" 'retain ?tests t ?vars t ?parameters t
?currentMode t
    ?runOptions t ?specs t ?corners t ?extensions t
    ?modelGroups nil ?relxanalysis nil )

runid2 = ocnxlRun(?waitUntilDone nil)
```

When the *waitUntilDone* argument is set to `nil`, OCEAN XL does not wait for the run to complete. Instead, while the run is in progress, it executes the next line in the script. In the example given above, OCEAN XL loads another state, `Tran_State2`, and starts a simulation for that without waiting for the first simulation run to finish.

Note: You need to set the *waitUntilDone* argument for each OCEAN XL run that you want to run in parallel. By default, this argument is set to `t`.

When you set the *waitUntilDone* argument to `nil`, the ocnxlRun function also returns a run ID for each run. You can use this run ID later to either wait for that run to complete at a later stage or to access the history results of that run.

For example, before printing results, if you want to wait for a particular OCEAN XL run to complete simulations, you can use the following statement in your script:

```
(ocnxlWaitUntilDone rinid2)
(ocnxlOutputSummary ?forRun runid1 ?detailed nil)
; The previous command displays run summary for the simulation run with run ID as
runid1
```

After a run is complete, you can get access to its results by using its run ID, as shown in the following statement:

```
h1 = ocnxlGetHistory(runid1)
```

You can use this handle to get results for the given OCEAN XL run.

For details on the related OCEAN XL functions and their examples, refer to the following sections in the OCEAN Reference Guide:

- ocnxlRun

- ocnxlOutputSummary
- ocnxlWaitUntilDone
- ocnxlGetHistory
- ocnxlSetRunDistributeOptions
- ocnxlGetRunDistributeOptions

Viewing Results of Simulations Run using OCEAN Scripts

The results of simulations run from OCEAN scripts are saved in the results database of the ADE XL view. While the OCEAN XL simulations are in progress or after they are complete, you can view these results in the ADE XL user interface.

To view the results for an OCEAN simulation run in the ADE XL user interface, do the following:

➡ In the History tab of the `Data View` assistant pane, right-click on the history item for the OCEAN run and choose *View Results*.

Note: While the simulations are in progress, you need to close and reopen the results to view the updated data.

Important

When you view the results of simulations performed by specifying netlist as design, you cannot use the schematic-based post processing options. For example, you cannot use the `VT` calculator function to select a net on schematic or you cannot use the direct plotting feature. Therefore, when you right-click on data in the *Results* tab, some of the commands that require use of the schematic view are not enabled.

Simulating Designs with Layout-Dependent Effects (LDEs)

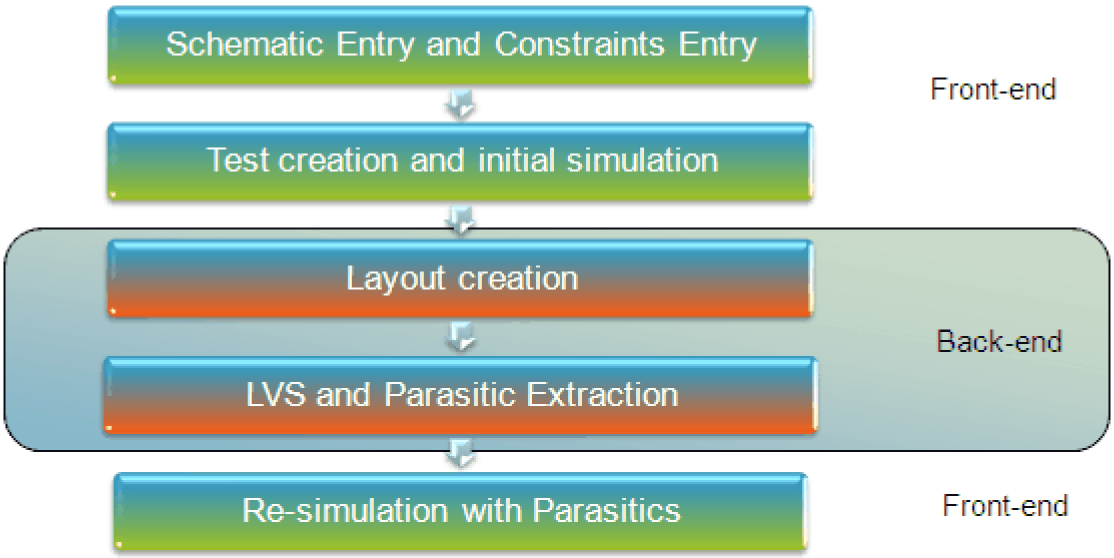
Important

Ensure that the following prerequisites are met before you run simulations with Layout-Dependent Effects (LDEs):

- ❑ The **Virtuoso_Variation_Analysis_Op** license is available for use. For more details, see the *Virtuoso Software Licensing and Configuration Guide*.
- ❑ Verify that the path to the Cadence Physical Verification System (PVS) installation is correct. The LDE re-simulation flow works with 10.1 or higher versions of PVS.
- ❑ Specify the path to the PVS-LVS rule file and layer map file by using the `lvsRulesFile` and `layerMapFile` environment variables, respectively. These files are required when performing LVS checks.

At advanced nodes, cells are placed very close together and such proximity can lead to various interconnect and layout-dependent effects. These LDEs, such as Shallow Trench Isolation (STI) and Well Proximity Effect (WPE), have substantial impact (in some cases, upto 30 percent difference) on the performance of a circuit. Therefore, it becomes very important to take into account these LDEs while simulating a design to measure the performance of a circuit accurately.

In the previous design flows, circuit designers used to take only an LVS-clean design to analyze the impact of layout decisions on the circuit performance, as described in the flow shown below.

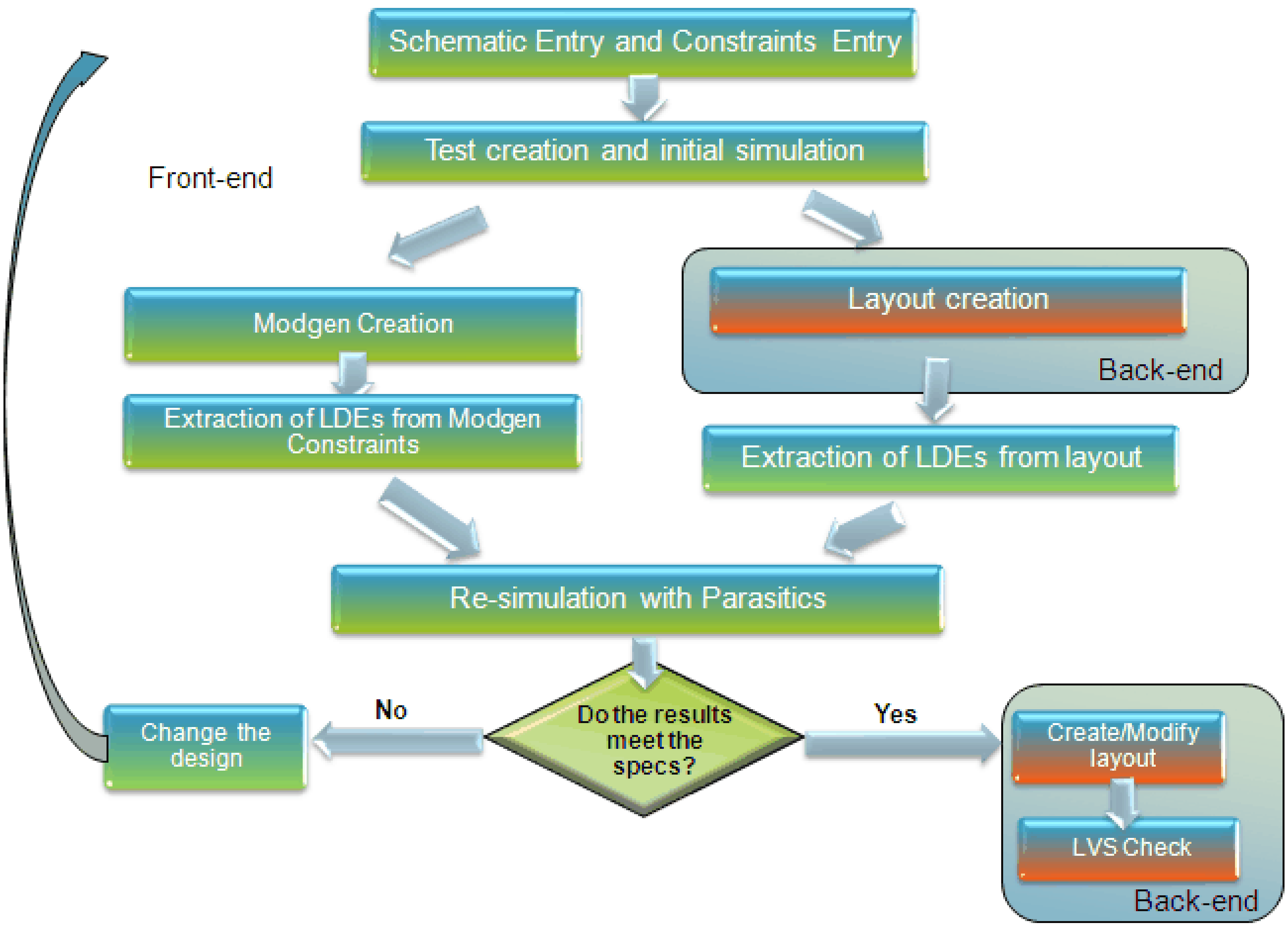


Re-simulation Flow in Earlier Releases

In advanced node designs, LDEs can have high impact on the circuit performance. Therefore, it is important to take them into account quite early in the design cycle, that is, while simulating a design before the completion or verification of the layout. This helps in avoiding too many design iterations because you can correct a design before finalizing the layout.

In the LDE re-simulation flow, you can use a partial or a full layout and extract more accurate LDE parameter values from it to be used in ADE XL simulations. This helps in simulating a circuit with a better model of the physical implementation.

ADE XL provides options to include extracted LDE parameter values in a simulation run. The LDE re-simulation flow is shown in the figure given below.



Design Flow for Advanced Manufacturing Node Designs

As depicted in the figure shown above, you can include LDE parameter values in ADE XL simulations in the following two ways:

- By including LDEs extracted from the Modgen constraint definitions extracted from the schematic
- By including LDEs extracted from a partial or a full layout

Simulating Designs with LDEs Extracted from Modgen Constraints

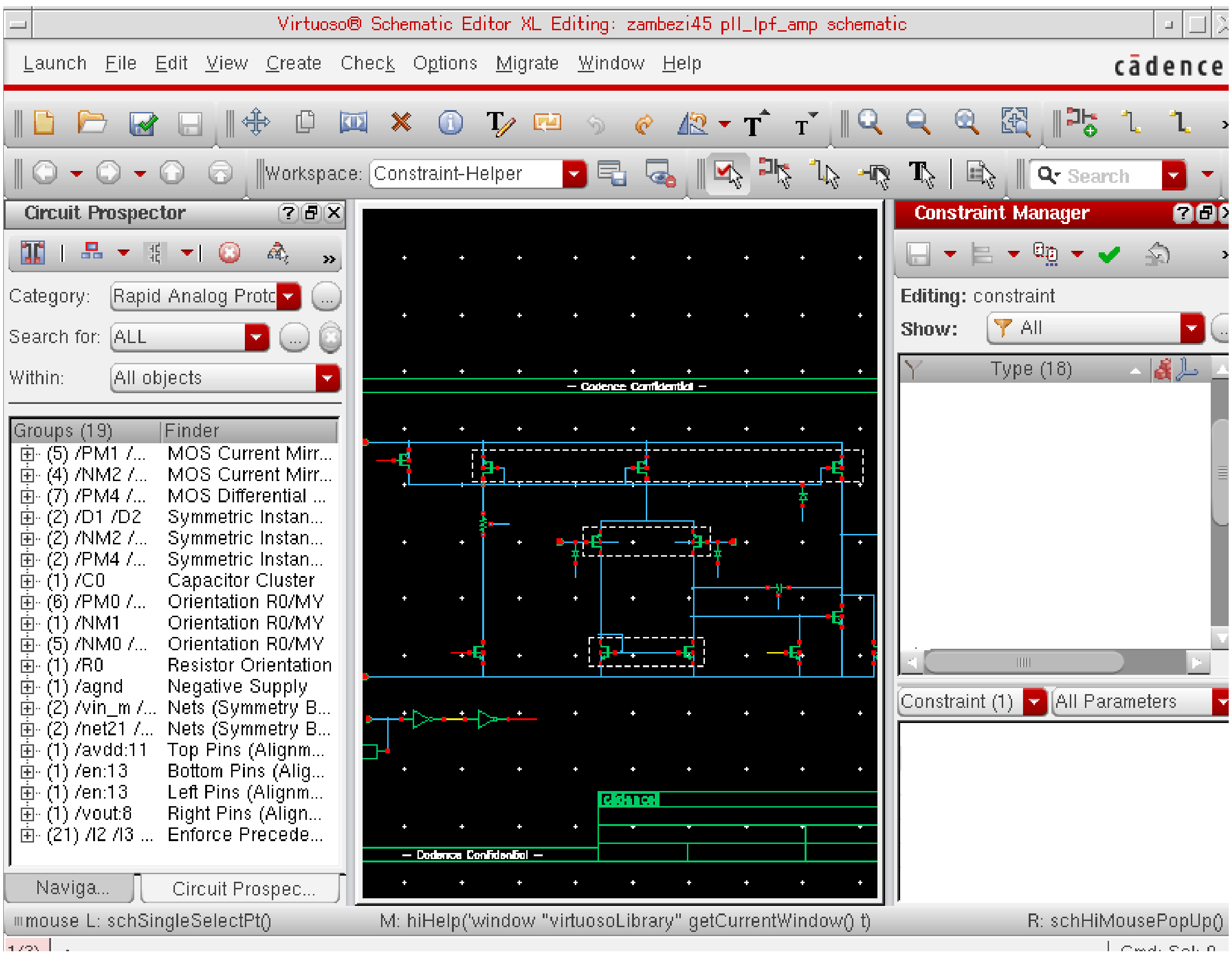
If a layout view is not available for your design, you can use design constraints to create optimized Modgen constraint definitions. Then, you can extract LDE parameter values from these Modgen constraint definitions and include them in the simulation netlist to analyze their effect on simulation results.

To create Modgen constraint definitions and to use extracted LDE parameter values from these Modgen definitions in simulations, perform the steps given below:

1. Create a simulation setup in an adexl view. For more details, refer to Specifying Tests and Analyses.
2. Run a simulation.

   Note: You have run a simulation without considering any parasitics or layout effects.
3. Open the design schematic in Virtuoso Schematic Editor XL.
4. Change the workspace to `Constraint-Helper`.

   The Circuit Prospector and Constraint Manager assistants are displayed.



5. Ensure that in the Circuit Prospector assistant, `Rapid Analog Prototype`, is selected from the *Category* drop-down list.

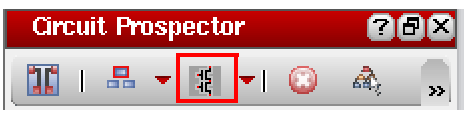
`Rapid Analog Prototype` is a new category in Circuit Prospector. This category contains many of the more common mixed-signal circuit finders and constraint generators required to constrain a design automatically. The resulting constraints can be used to create good initial layouts and routes from which layout effects (interconnect parasitic and LDEs) can be extracted and taken into account during simulation. However, note that this flow does not extract interconnect parasitics.

For more details on this category, refer to The Circuit Prospector Assistant in *Virtuoso Unified Custom Constraints User Guide*.

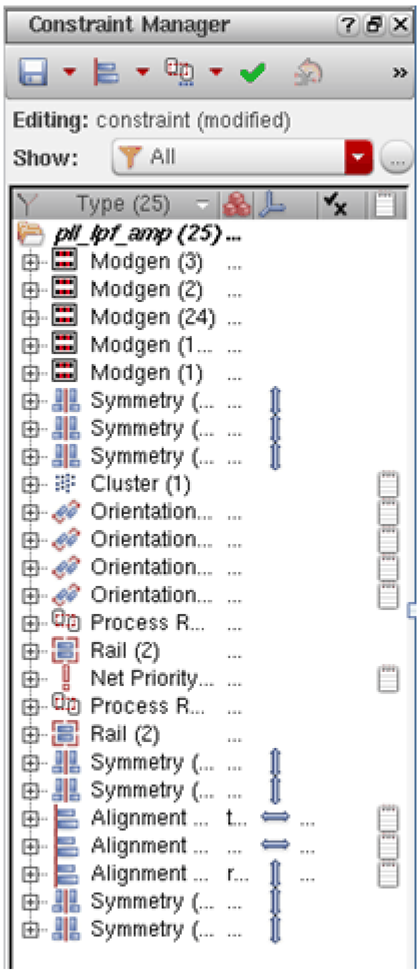
6. From the *Search for* drop-down list, select `ALL`.

   All the structures defined in this category are found in the schematic, if they exist.

7. Click *Run all respective actions for the Current Cellview* on the Constraints menu.

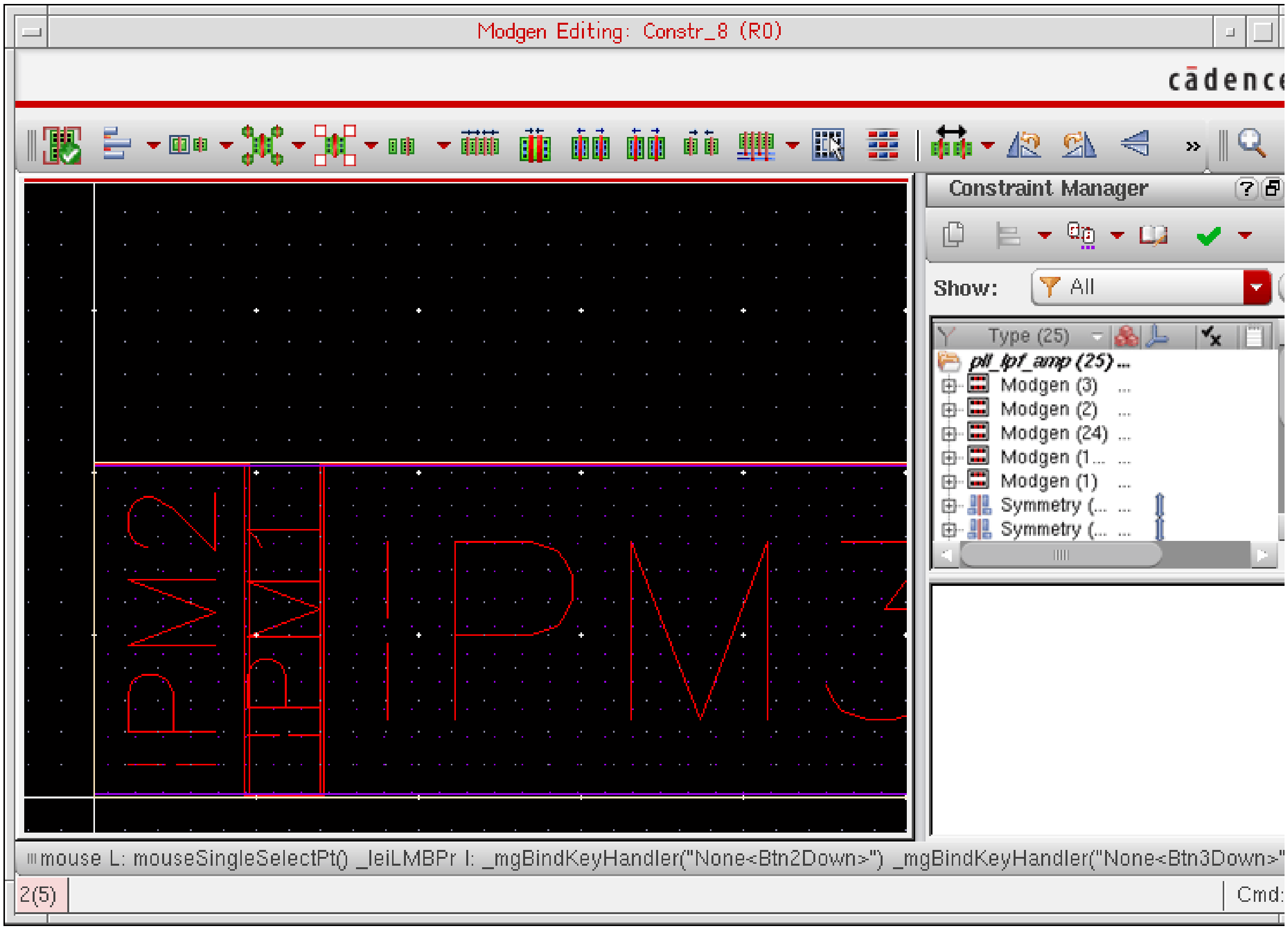


8. The tool runs the default constraint generator for all the structures identified by the `Rapid Analog Prototype` filter. All the constraints are populated in the Constraint Manager assistant, as shown below.



Note: For more details on how the constraints are generated, refer to <u>RAP Flow</u> in *Virtuoso Unified Custom Constraints User Guide*.

9. (Optional) If required, you can open a Modgen constraint in the Modgen Editing tool and make changes in the prototype layout, for example, you can change the abutment for modules or add a guard ring or dummy cells.



For more details, refer to Working with Module Generators in Virtuoso Analog Placer User Guide.

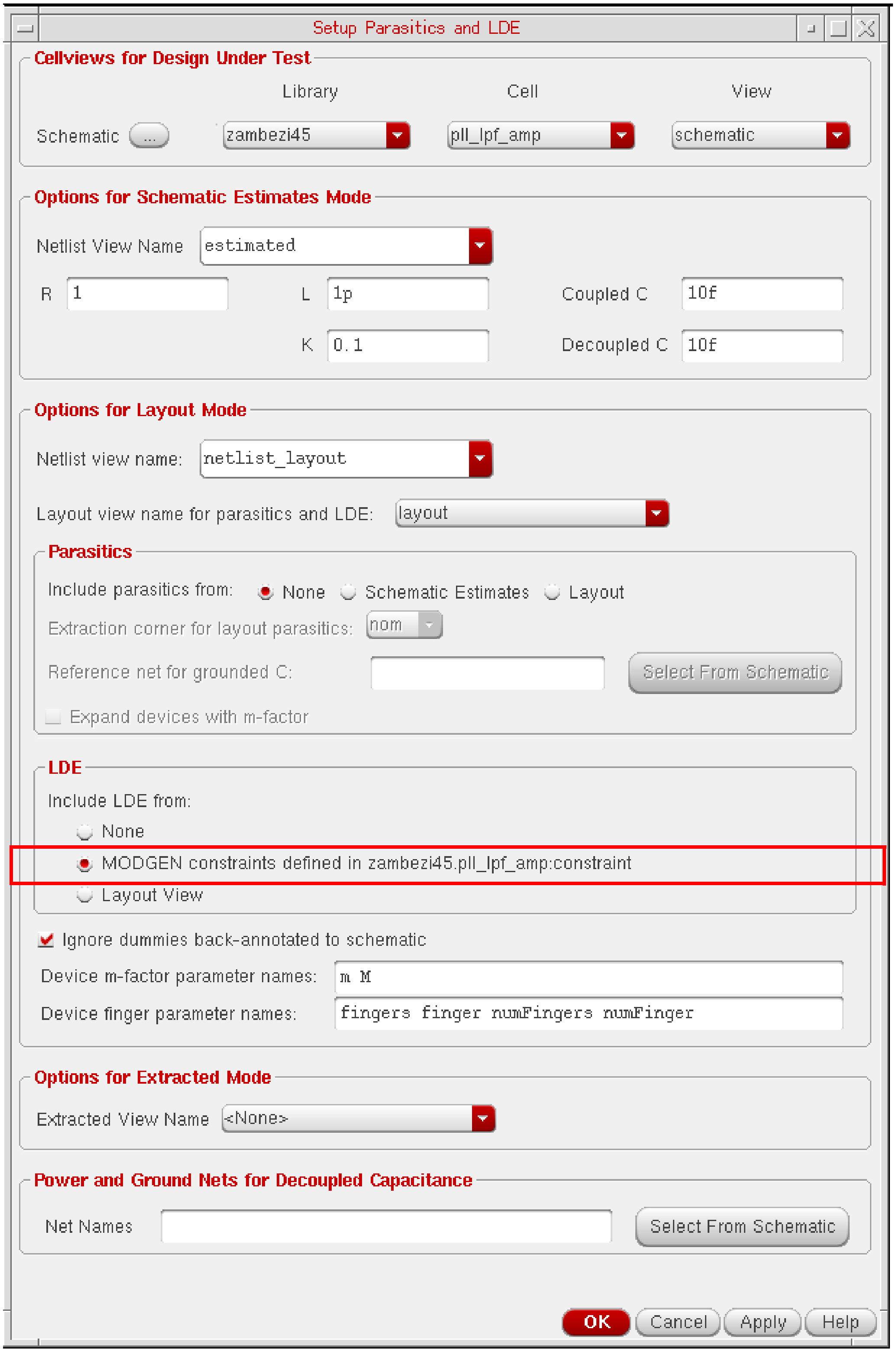
10. Open the same adexl view that you used in step 1.

11. In the Analog Design Environment XL window, choose *Parasitics/LDE – Setup*.

The Setup Parasitics and LDE form is displayed.

12. Specify a netlist view name in the *Netlist view name* field.

13. In the *Options for Layout Mode* section, select the *MODGEN Constraints defined in <constaint-view-name>* option for *Include LDE from*, as shown below.


Setup Parasitics and LDE
Cellviews for Design Under Test
Library
Cell
View
Schematic
...
zambezi45
pll_lpf_amp
schematic
Options for Schematic Estimates Mode
Netlist View Name
estimated
R
1
L
1p
Coupled C
10f
K
0.1
Decoupled C
10f
Options for Layout Mode
Netlist view name:
netlist_layout
Layout view name for parasitics and LDE:
layout
Parasitics
Include parasitics from:
None
Schematic Estimates
Layout
Extraction corner for layout parasitics:
nom
Reference net for grounded C:
Select From Schematic
Expand devices with m-factor
LDE
Include LDE from:
None
MODGEN constraints defined in zambezi45.pll_lpf_amp:constraint
Layout View
Ignore dummies back-annotated to schematic
Device m-factor parameter names:
m M
Device finger parameter names:
fingers finger numFingers numFinger
Options for Extracted Mode
Extracted View Name
<None>
Power and Ground Nets for Decoupled Capacitance
Net Names
Select From Schematic
OK
Cancel
Apply
Help

14. (Optional) In the *Device m-factor parameter names* field, specify the names of CDF parameters that define m-factors for devices.

    For details on how to define CDF parameters, refer to *Component Description Format User Guide*.

15. (Optional) In the *Device finger parameter names* field, specify the names of CDF parameters that define fingers for devices.

16. Click *Apply*.

    If the parasitic mode on the Parasitic Mode toolbar is not set to `Layout(LDE/Parasitics)`, ADE XL prompts you to confirm if the mode is to be changed.

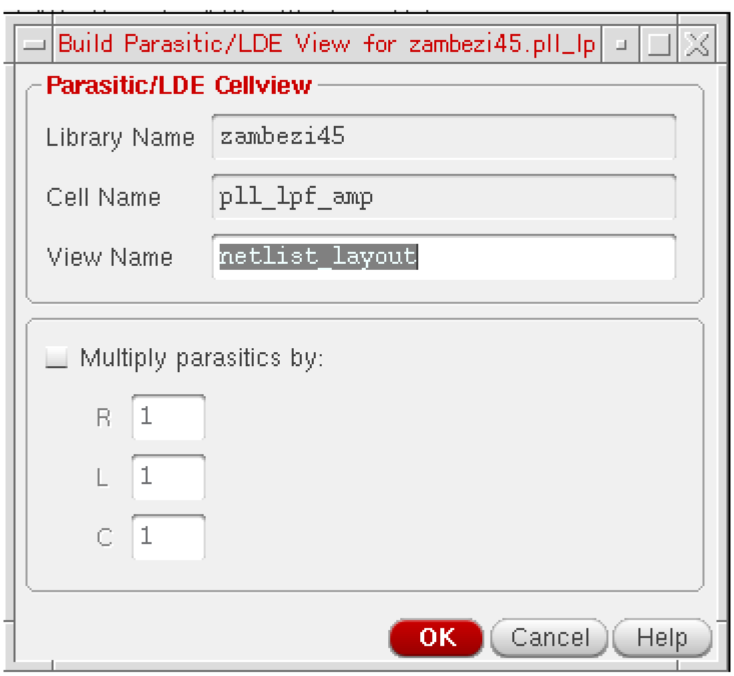
    The tool also prompts you to build the estimated (`netlist_layout`) view with LDE parameters. You can choose to build the view estimated view now or from the Parasitic Mode toolbar after closing the form.

17. Click *OK*.

18. Ensure that the *Sweeps* drop-down list on the Parasitic Mode toolbar is set to `Devices Only` to sweep only devices.

19. Click *Build Parasitic/LDE View* on the Parasitic Mode toolbar.

    The Build Parasitic/LDE View form is displayed, as shown below.



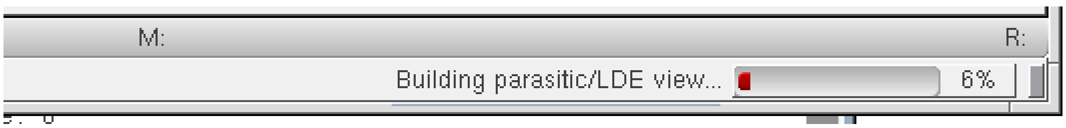
20. (Optional) Specify a name for the parasitic/LDE view.

Note: `netlist_layout` is the default name for the parasitic/LDE view.

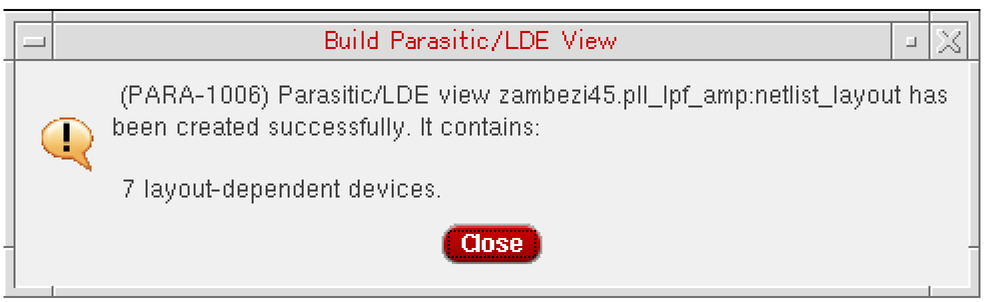
21. Click *OK* to close the Build Parasitic/LDE View form.

ADE XL runs PVS-LVS, which reads the Modgen constraints and extracts the values for LDE parameters to create a new netlist. It further creates a parasitic/LDE view.

While the parasitic/LDE view is being built, the ADE XL status bar shows the progress.



After completion, the tool shows a pop-up message with the number of devices defined in the Modgen constraints. A message is also displayed in the CIW.



Note: Ensure that the paths to the PVS-LVS rule file and layer map file are specified by using the *lvsRulesFile* and *layerMapFile* environment variables. PVS refers to these files while performing the LVS check.

22. Click Run to re-simulate the ADE XL testcase.

23. Compare the simulation results of this run with the history of the previous run and identify the effect of LDEs on circuit performance.

If the circuit performance has deteriorated or needs improvement, you can choose to modify the constraints by using the Schematic Editor, or change the layout by using Modgen Editor, regenerate the LDE parameter estimated view and run simulations again to measure the results.

Note: When simulating designs with LDEs, Virtuoso runs PVS-LVS to read the Modgen constraints and to extract the values of LDE parameters. The Virtuoso IPVS log file is saved in the `/tmp/<username>_pvs_*` directory and is automatically deleted when you exit the Virtuoso session. However, for debugging purposes, you can preserve the Virtuoso IPVS log file. For this, set the *Virtuoso_IPVS_log* environment variable before starting Virtuoso. For more details, refer to Debugging Problems in Virtuoso IPVS in the *Virtuoso IPVS User Guide*.

For more details on the Setup Parasitics and LDE form options, refer to Setting Up and Using Parasitics in *Parasitics Aware User Guide*.

Simulating Designs with LDEs Extracted from a Partial or Full Layout

ADE XL simulation with LDE parameters can also be done by using the partial layout capture, which would not be a LVS clean layout, or a full LVS clean layout.

Note: A layout, partial or full, is created by using the connectivity-driven layout flow of Virtuoso Layout Suite XL.

To simulate a design by using the LDEs extracted from a layout view, perform the following steps:

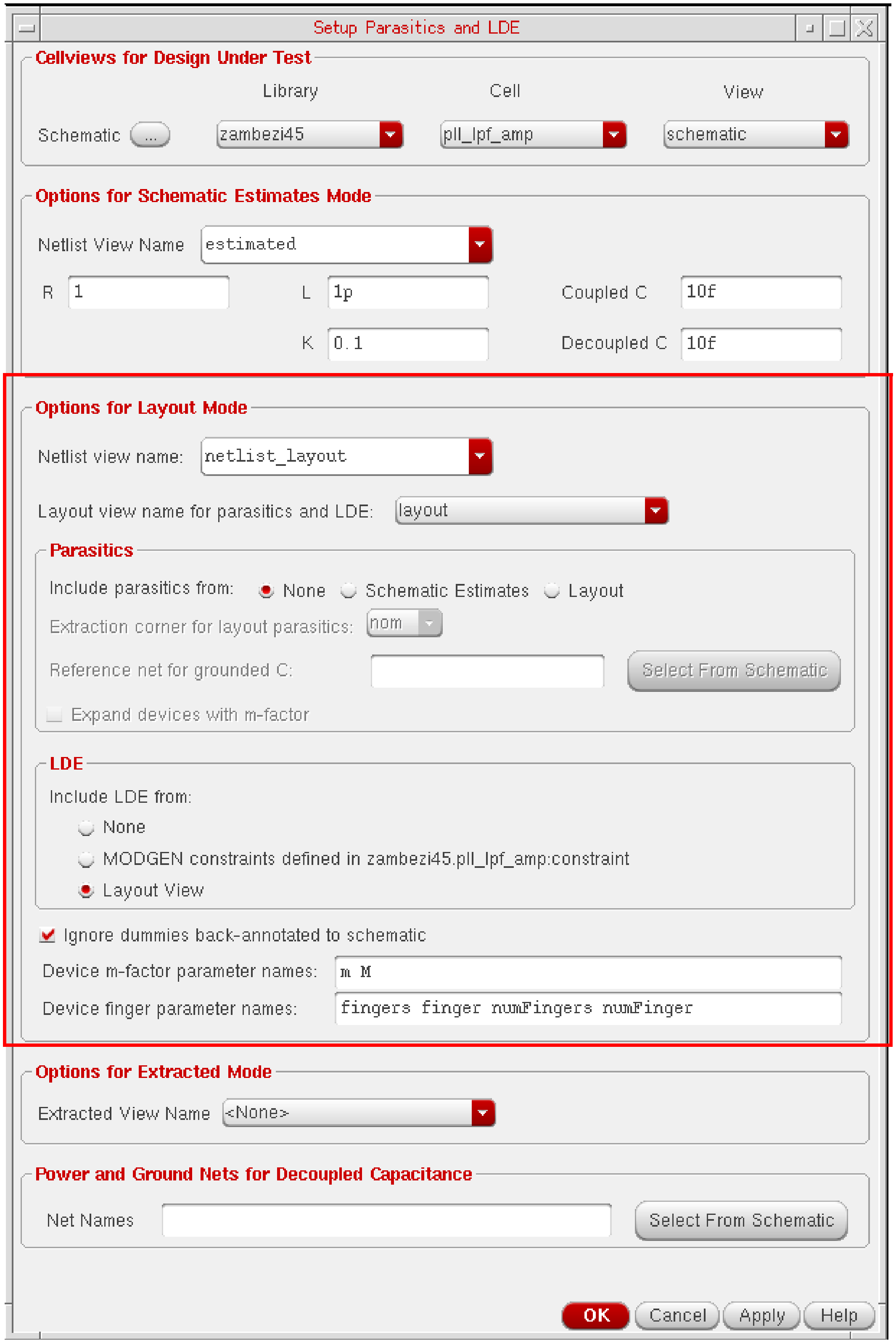
1. In the Analog Design Environment XL window, choose *Parasitics/LDE – Setup*.

   The Setup Parasitics and LDE form is displayed.

2. In the *Options for Layout Mode* section, specify a name for the netlist view in the *Netlist View Name* field.

3. In the *Layout view name for parasitics and LDE* list, select the names of the layout view, which contains LDE parameters.

4. Select *Layout View* for the *Include LDE From* field, as shown below.


Setup Parasitics and LDE
Cellviews for Design Under Test
Library
Cell
View
Schematic
zambezi45
pll_lpf_amp
schematic
Options for Schematic Estimates Mode
Netlist View Name
estimated
R
1
L
1p
Coupled C
10f
K
0.1
Decoupled C
10f
Options for Layout Mode
Netlist view name:
netlist_layout
Layout view name for parasitics and LDE:
layout
Parasitics
Include parasitics from:
None
Schematic Estimates
Layout
Extraction corner for layout parasitics:
nom
Reference net for grounded C:
Select From Schematic
Expand devices with m-factor
LDE
Include LDE from:
None
MODGEN constraints defined in zambezi45.pll_lpf_amp:constraint
Layout View
Ignore dummies back-annotated to schematic
Device m-factor parameter names:
m M
Device finger parameter names:
fingers finger numFingers numFinger
Options for Extracted Mode
Extracted View Name
<None>
Power and Ground Nets for Decoupled Capacitance
Net Names
Select From Schematic
OK
Cancel
Apply
Help

Note: The *Expand devices with m-factor* check box is disabled when you select the option to include LDE parameters from a layout view. This is because in this mode, the tool always considers the m-factor parameters from the layout view to expand the devices in the netlist_layout view.

5. (Optional) In the *Device m-factor parameter names* field, specify the names of CDF parameters that define m-factors for devices.

   For details on how to define CDF parameters, refer to *Component Description Format User Guide*.

6. (Optional) In the *Device finger parameter names* field, specify the names of CDF parameters that define fingers for devices.

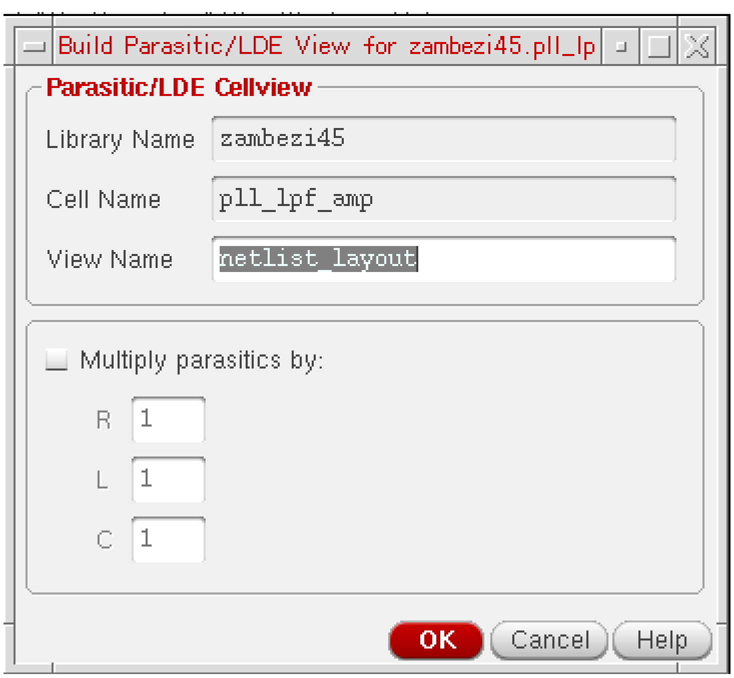
7. By default, the dummy cells back annotated from layout to the schematic view are ignored while generating netlist for simulation. To include the dummy cells in the netlist, clear the *Ignore dummy devices back-annotated to schematic* check box.

8. Click *OK* to close the form.

9. On the Parasitic Mode toolbar, select the `Layout(LDE)` parasitic mode and `Devices Only` to sweep only devices.

10. Click *Build Parasitic/LDE View* on the Parasitic Mode toolbar.

    The Build Parasitic/LDE View form is displayed, as shown below.



11. (Optional) Specify a name for the parasitic/LDE view. By default, it shows the value specified in the Setup Parasitics and LDE form.

12. Click *OK* to close the Build Parasitic/LDE View form.

ADE XL runs PVS-LVS, which reads the design layout, extracts the values for the LDE parameters of the devices captured in the layout and stitches that information in the netlist along with the instances of the non-laid-out devices in the schematic.

13. (Optional) To include the device parameters while running simulations, ensure that the parameters list is defined in the Data View assistant and the *Parameters* check box is selected.

14. Click Run to re-simulate the adexl view and compare results.

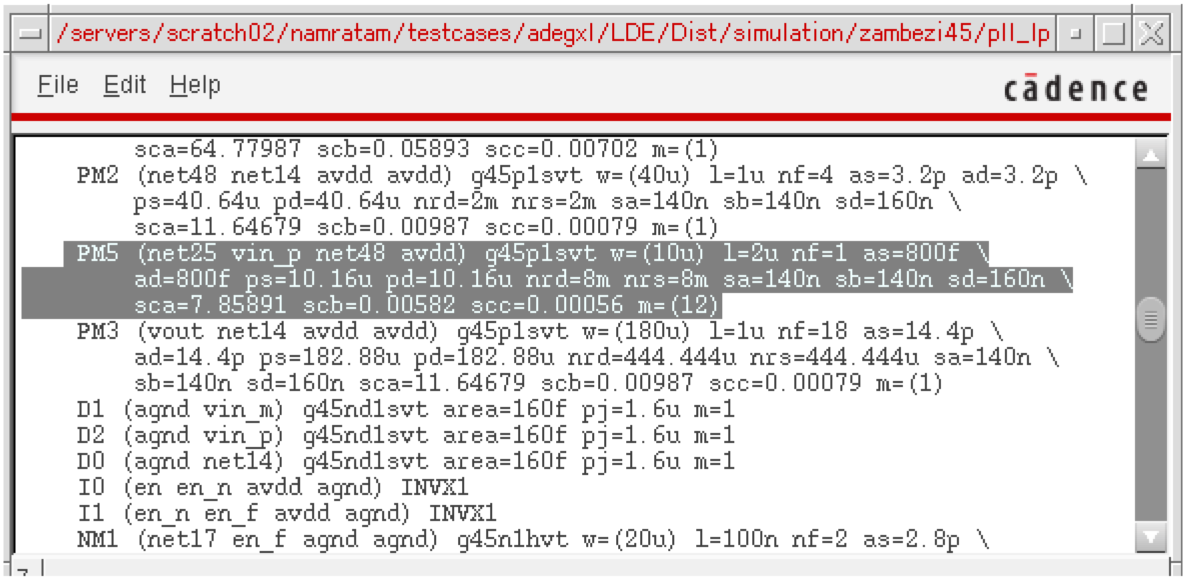
Important Points to Note

- You can run the Local Optimization or Global Optimization run modes to identify the best design points that meet the desired specifications. The optimization flow will consider all the device parameters defined in the schematic, except those that define the m-factor for devices. For more details, refer to Optimizing an Extracted or Layout View in the *Virtuoso Parasitic Aware Design Flow User Guide*.
- If the simulation results need improvement, you can change the LDE parameters and rerun the simulation in ADE XL to evaluate the performance change.

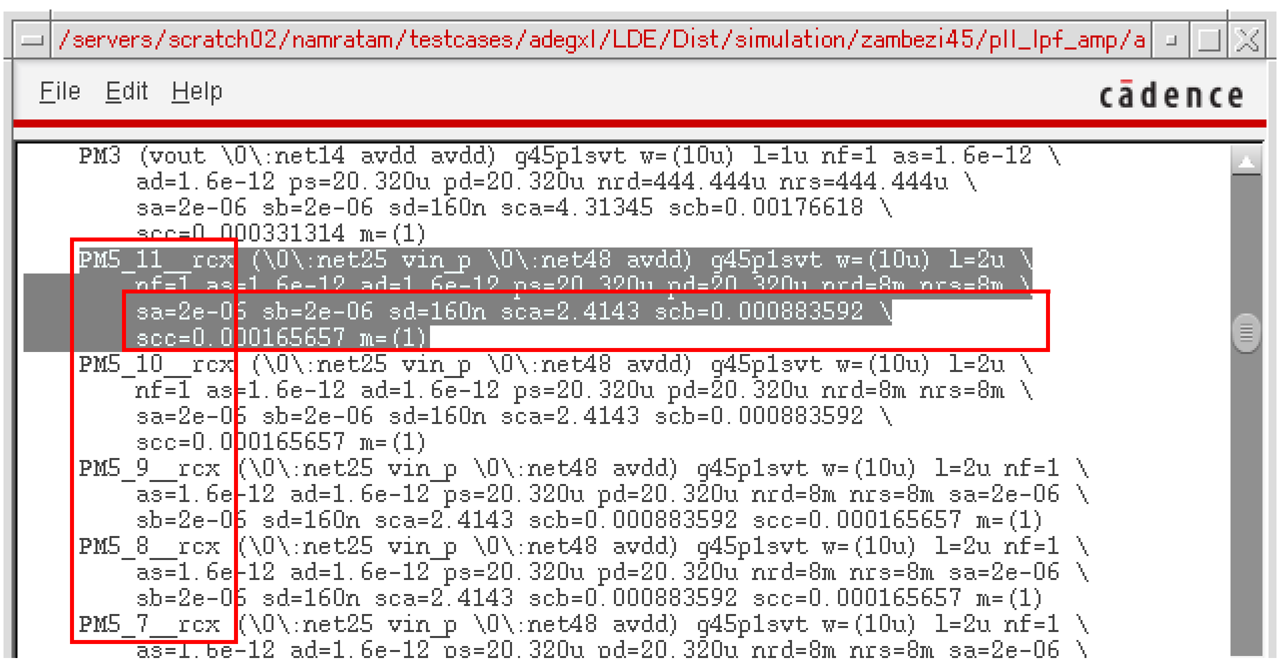
Additional Information

If you compare the netlists created before and after extracting the LDE parameters, you can observe that the new netlist created with the updated LDE parameter Parasitic/LDE view contains different values for these parameters as compared to the netlist created without parasitics.

The figure given below shows the netlist created without parasitic effects.



The figure given below shows the netlist created with extracted LDE parameters.



Note that for device PM5, the values of LDE parameters, such as `sa`, `sb`, `sca`, and `scb` are different. In addition, for the devices with m-factor in the layout view, multiple instances have been created. For example, multiple instances have been created for PM5 in the netlist shown above.

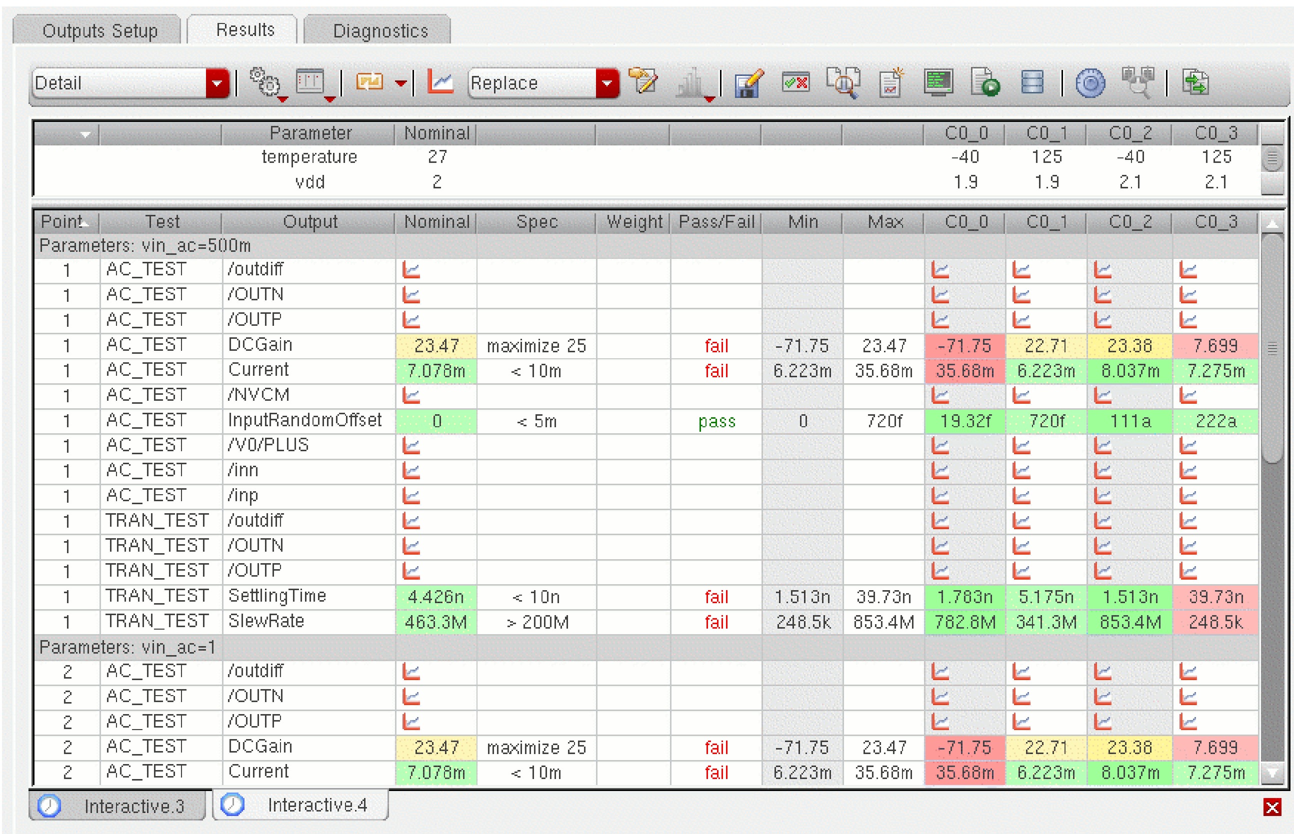
Tip

You can run Litho/LDE Analysis from Virtuoso Layout Suite XL to view violations and to debug the layout. For more details, refer to Litho/LDE Analysis in the *Virtuoso Layout XL User Guide*.

14

Viewing, Plotting, and Printing Results

You can view simulation results on the Results tab of the Outputs pane in the ADE XL environment.



Results appear on a new tab in the Results tab of the Outputs pane after you run a successful simulation, or view or restore results from a previous checkpoint on the History tab of the Data View assistant pane. For more information about working with tabs, see Working with Tabs for Simulation Checkpoints on page 561. For more information about how results are displayed in the Results tab, see Viewing Specification Results in the Results Tab on page 737.

By default, all columns and types of information appear on the Results tab. You can show and hide various columns and types of information using the  button. Corners conditions appear in a table along the top portion of the tab. You can switch the view using the *Select the results view* drop-down list. You can export results to HTML or a comma-separated values file.

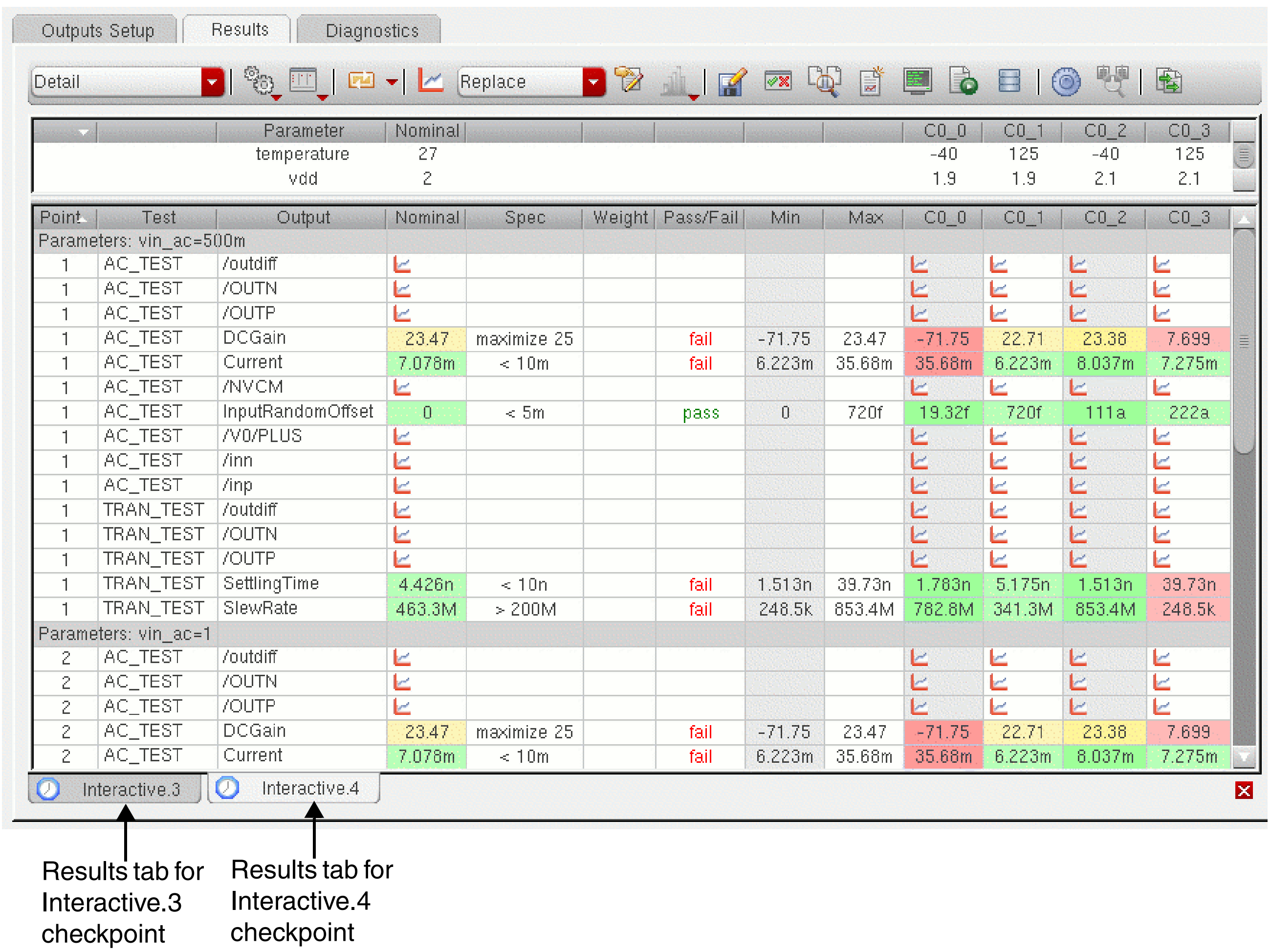
See the following topics for more information:

See also "Results Tab Right-Click Menus" on page 699.

Working with Tabs for Simulation Checkpoints

When you run a simulation, or view or restore results from a previous checkpoint on the History tab of the Data View assistant pane, the simulation results appear on a new tab in the Results tab of the Outputs pane. The name of the tab matches the name of the checkpoint for the simulation run. If you view or restore results from a previous checkpoint, the name of the tab matches the name of that checkpoint.

For example, in the following figure, the *Interactive.4* tab displays the results for the `Interactive.4` checkpoint. To view the results for the `Interactive.3` checkpoint, click the *Interactive.3* tab.



The icons that are displayed in the tab name when you run a simulation are described below:

Icon	Description
	Simulation run for `Interactive.16` checkpoint is started.
	Simulation run for `Interactive.16` checkpoint is in progress. The progress of the simulation run is displayed in the form of a pie chart.
	Simulation run for `Interactive.16` checkpoint is complete.

To close the currently active tab, do the following:

➡ Click the  icon on the bottom right of the Results tab.

To close a specific tab, do the following:

➤ Right-click on the tab name and choose *Close Tab*.

To close all tabs other than the current one, do the following:

➤ Right-click on the tab name and choose *Close Other Tab*s.

To close all tabs, do the following:

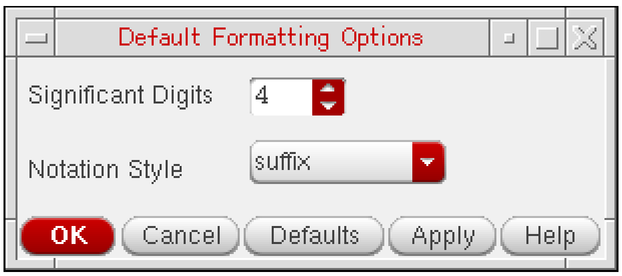
➤ Right-click on the tab name and choose *Close All Tabs*.

Specifying Default Formatting Options

To specify the default number of significant digits and notation style for the measured results displayed in the Results tab of the Outputs pane, do the following:

1. Choose *Options – Outputs Formatting*.

   The Default Formatting Options form appears.



2. Specify the number of significant digits in the *Significant Digits* field.

   The default value is `4`. Valid values are `2` to `15`.

3. Select the notation style in the *Notation Style* drop-down list.

Select	To
suffix	Display the results in the suffix notation. For example, `5.802m` is displayed as `5.802m` when number of significant digits is `4`. This is the default value.
engineering	Display the results in the engineering notation. For example, `5.802m` is displayed as `5.802e-3` when number of significant digits is `4`.
scientific	Display the results in the scientific notation. For example, `5.802m` is displayed as `5.802e-03` when number of significant digits is `4`.

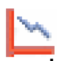
Tip

You can use the *Units*, *Digits*, *Notation* and *Suffix* columns in the Outputs Setup tab of the Outputs pane to override the default number of significant digits and notation style for displaying the results for individual output expressions. For more information, see Adding an Output Expression on page 376.

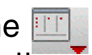
Hiding and Showing Data on the Results Tab

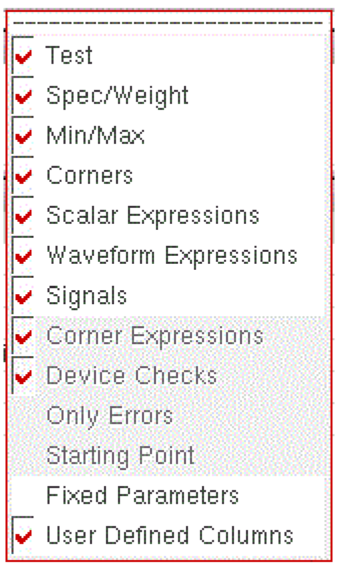
The following data appears on the Results tab of the Outputs pane for the Detail view. See "About Results Views" on page 572 for more information about other views in the Results tab.

Table 14-1 Columns in the Detail View

Column	Data/Content	Hide/Unhide
Point	Number of the design point for which results are displayed.	
Test	Test name, top of expandable tree of outputs	See "Hiding Test Details" on page 566
Output	Signal or expression/measurement name	
Nominal	Plot  icon for signals and waveform expressions; status indicator box and measured value for expressions at the nominal corner: green indicates a value within the specifications; yellow indicates a value that is no more than 10% outside the target value of specifications; red indicates a value that is more than 10% outside the target value	See "Hiding Signals" on page 566 See "Hiding Measured Result Values" on page 567
Spec	Specification information	See "Hiding Specification Details" on page 566
Weight	Weighting factor given to a specification	
Pass/Fail	*pass* when values are within specifications; *near* when one or more measured values are no more than 10% outside the target value of the specification; *fail* when one or more measured values are more than 10% outside the target value of the specification For more information about the *pass*, *near* or *fail* status, see Viewing Specification Results in the Results Tab on page 737.	
Min	Minimum value for the specification	"Hiding Minimum and Maximum Values" on page 567
Max	Maximum value for the specification	

Column	Data/Content	Hide/Unhide
CornersName	Measured value or waveform at each corner	See "Hiding Corner Results" on page 566

You can show or hide columns and rows displayed on the Results tab by using the  (*Configure what is shown in the table*) button on this tab. When you click this button, ADE XL shows the following menu.



For information about these options, see the following topics:

- Hiding Test Details on page 566
- Hiding Specification Details on page 566
- Hiding Corner Results on page 566
- Hiding Signals on page 566
- Hiding Signals on page 566
- Hiding Measured Result Values on page 567
- Hiding Minimum and Maximum Values on page 567
- Showing Only Errors on page 567
- Showing the Starting Point on page 568
- Showing Fixed Parameters for a Design Point on page 568

Hiding Test Details

Test details appear in the *Test* column on the Results tab of the Outputs pane. To hide test details, do the following:

➤ Click the (*Configure what is shown in the table*) button on the Results tab and choose *Test*.

The *Test* column disappears from the Results tab.

To unhide this column, click the button once again and choose *Test*.

Hiding Specification Details

Specification details appear in the *Spec* and *Weight* columns on the Results tab of the Outputs pane. To hide specification details, do the following:

➤ Click the button on the Results tab and choose *Spec/Weight*.

The *Spec* and *Weight* columns disappear from the Results tab.

To unhide these columns, click the button once again and choose *Spec/Weight*.

Hiding Corner Results

Corner results appear in columns on the Results tab of the Outputs pane, each result in its own column. To hide corner results columns, do the following:

➤ Click the button on the Results tab and choose *Corners*.

The corner results columns disappear from the Results tab.

To unhide these columns, click the button once again and choose *Corners*.

Hiding Signals

Signals appear as rows with the icon in the *Nominal* column and the columns for corners on the Results tab of the Outputs pane.

To hide signals, do the following:

➤ Click the button on the Results tab and choose *Signals*.

The signal rows disappear from the Results tab. If you have any measured values for expressions, those are the only values that appear.

To unhide these rows, click the button once again and choose *Signals*.

Hiding Measured Result Values

Measured result values appear in the *Nominal* column and the columns for corners on the Results tab of the Outputs pane.

To hide measured result values, do the following:

➤ Click the button on the Results tab and choose *Expressions*.

The rows displaying measured result values disappear from the Results tab. The Results tab displays only the information about signals.

To unhide these rows, click the button once again and choose *Expressions*.

Hiding Minimum and Maximum Values

The minimum and maximum values appear in the *Min* and *Max* columns on the Results tab of the Outputs pane. To hide minimum and maximum values, do the following:

➤ Click the button on the Results tab and choose *Min/Max*.

The *Min* and *Max* columns disappear from the Results tab.

To unhide these columns, click the button once again and choose *Min/Max*.

Hiding Device Checks

To hide device checks, do the following:

➤ Click the button on the Results tab and choose *Device Checks*.

To unhide these rows, click the button once again and choose *Device Checks*.

Showing Only Errors

To show errors only, do the following:

➤ Click the button on the Results tab and choose *Only Errors*.

To hide these rows, click the button once again and choose *Only Errors*.

Showing the Starting Point

The reference point specified for Monte Carlo Sampling, Global Optimization, Local Optimization, Improve, Sensitivity Analysis, Feasibility Analysis and Size Over Corners runs is displayed as the starting point on the Results tab of the Outputs pane. To show the starting point on the top of the Results tab, do the following:

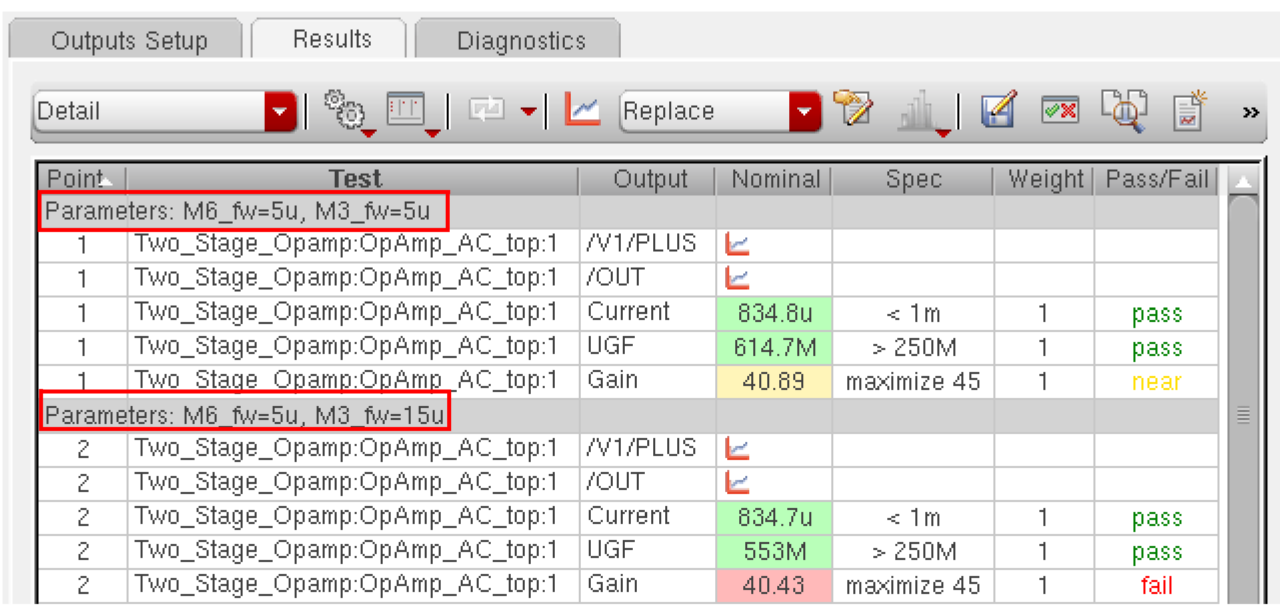
➤ Click the button on the Results tab and *Starting Point*.

The rows displaying the starting point appears on the top of the Results tab.

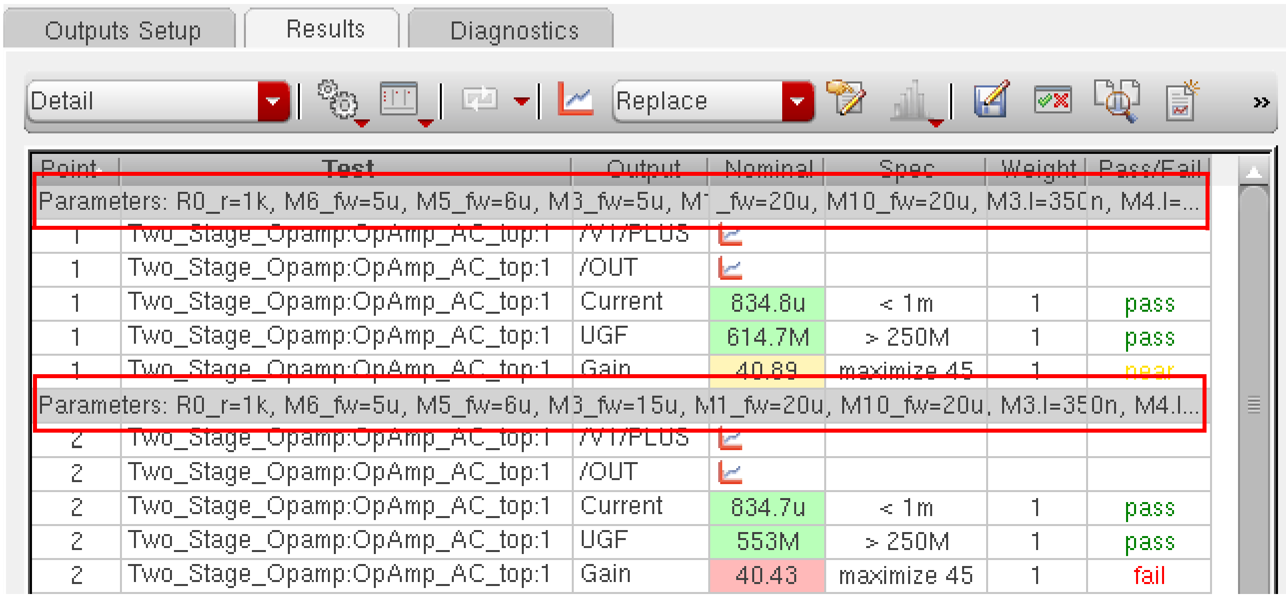
To hide these rows, click the button once again and choose *Starting Point*.

Showing Fixed Parameters for a Design Point

By default, for every design point, parameters names and their values are displayed in the gray bar on top of the results shown for that design point. For example, in the following figure, for each design point, the gray bars shows the values set for the two sweeped parameters.



If, in addition to the sweeped parameters, you want to display the fixed parameters on the gray bar, click the button on the Results tab and choose *Fixed Parameters*.



To hide the fixed parameters, click once again and choose *Fixed Parameters*.

Note: Similarly, you can show or hide fixed parameters on the Variable Display assistant.

Hiding and Showing Results for Tests

By default, the *Results* tab displays the results for all the tests. However, if you need to focus on the results of a particular test, you can choose to hide the rows related to that test.

To view only the results for a specific test, do the following:

1. Click the button on the Results tab and select *Hide All Tests*.
2. Click the button on the Results tab and select a specific test.

To view the results for all the tests, do the following:

➤ Click the button on the Results tab and select *Show All Tests*.

To hide the results for all tests, do the following:

➤ Click the button on the Results tab and select *Hide All Tests*.

The tool filters out the results for the selected tests.

Note that in case the tests use overridden values for variables, the parameter header rows are updated to show the variable values related to only those tests that are visible. For more details, refer to How Results are Displayed When A Global Variable is Disabled for a Test?.

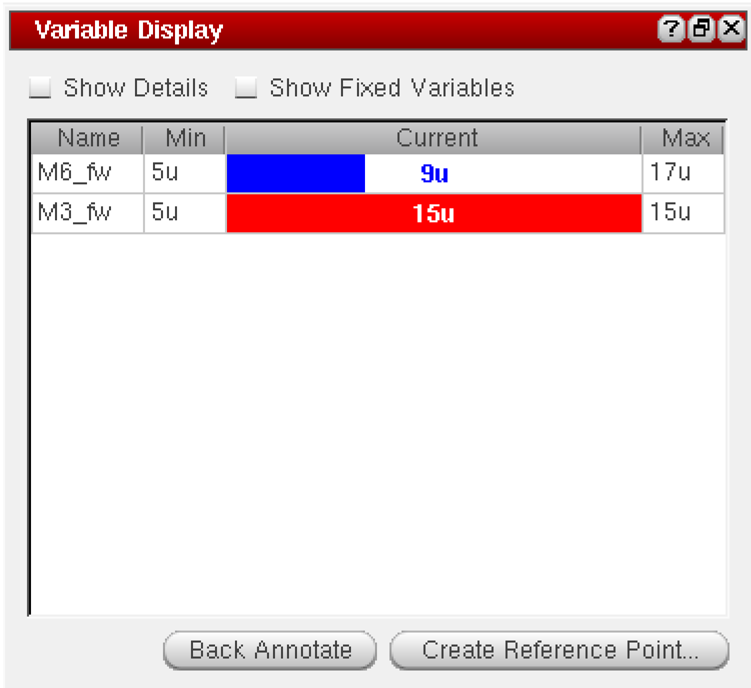
Similarly, if a corner is disabled for a test, only the corners that are enabled for the visible tests are shown. Columns that contain the results for the corners that are disabled for the visible tests are hidden. For more details, refer to Viewing Disabled Corners in the Results Tab.

Showing Variables for a Design Point

By default, the Results tab displays the names and values of variable parameters, which were used for a particular design point run, on the gray row above the results for a design point. If you also want to view the values of fixed parameters for a design point, click and choose *Fixed Parameters*.

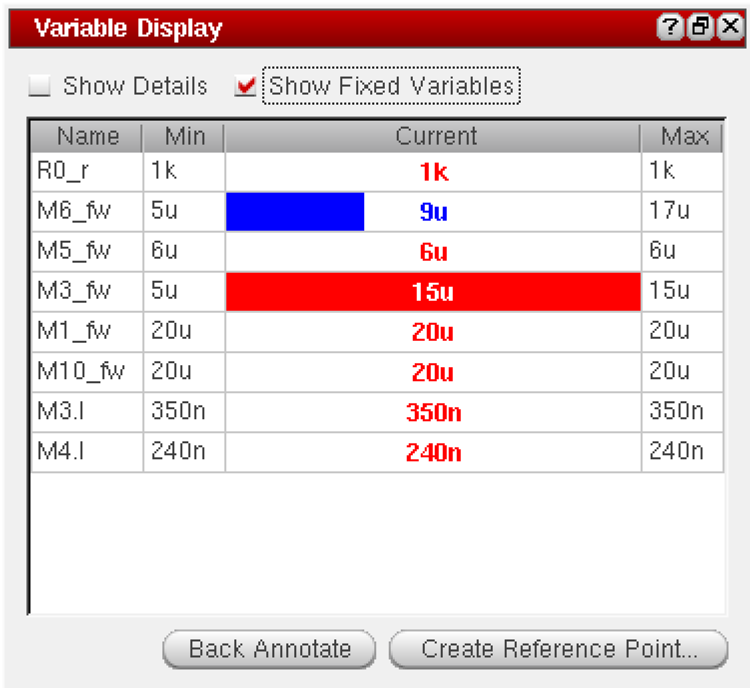
You can also view the list of all the variable parameters in the Variable Display assistant. To open this assistant, choose *Window – Assistant – Variable Display*. The Variable Display assistant is displayed at the bottom of the ADE XL window.

Click on any design point on the Results tab. A list of all the variables and their values used for a particular design point are displayed in the Variable Display assistant.



The Variable Display assistant shows the name of each variable with the minimum and maximum values and the current value used for the currently selected design point. The blue and red color bars indicate the sweep stage for a variable. For example, in the above figure, the blue bar for *M6_fw* shows that `9u` is one of the sweep values for this variable and there are few more sweeps. The red bar for *M3_fw* shows that `15u` is the maximum sweep value used for this variable.

By default, the fixed parameters are not shown in the Variable Display assistant. Click *Fixed Parameters* on top to show all fixed and variable parameters, as shown below.

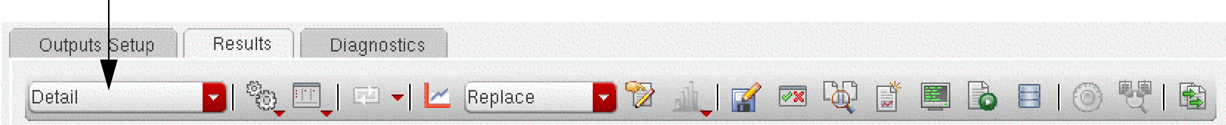


About Results Views

By default, the columns listed in Table 14-1 on page 564 are displayed in the Results tab of the Outputs pane. This view is called the *Detail* view. You can use the *Select the results view* drop-down list to switch between the following views:

- Detail
- Detail - Transpose
- Optimization
- Status
- Summary
- Yield

Select the results view
drop-down list



Every run mode has a default result view mapped to it. For example, the default result view for the Monte Carlo Sampling run mode is the Yield view.

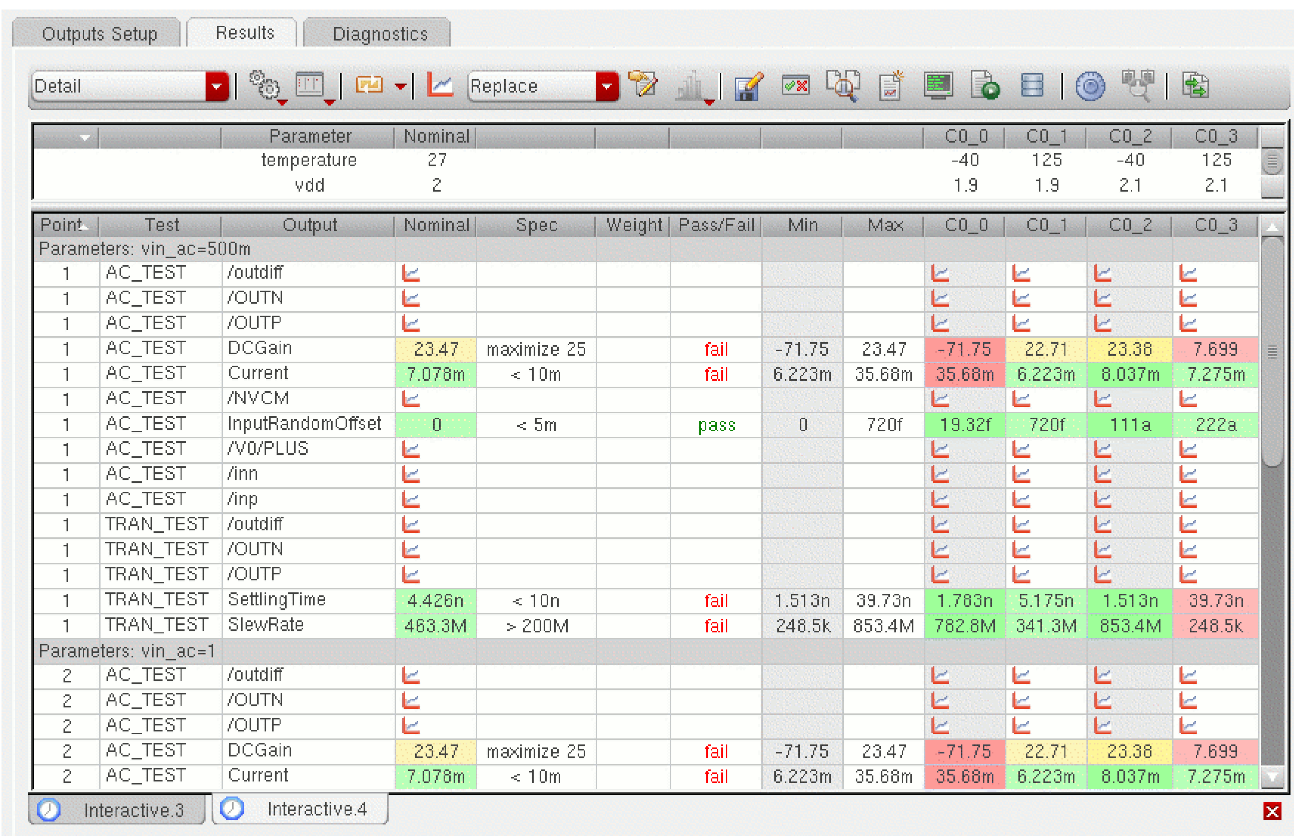
Note: You can change the default results view for the Single Run, Sweeps and Corners run mode by using the defaultResultsViewForSweepsCorners on page 898environment variable. This feature is currently not available for other run modes.

Detail

Displays the detailed results.

For more information about the columns in the *Detail* view, see Table 14-1 on page 564.

Figure 14-1 Detail View in Results Tab

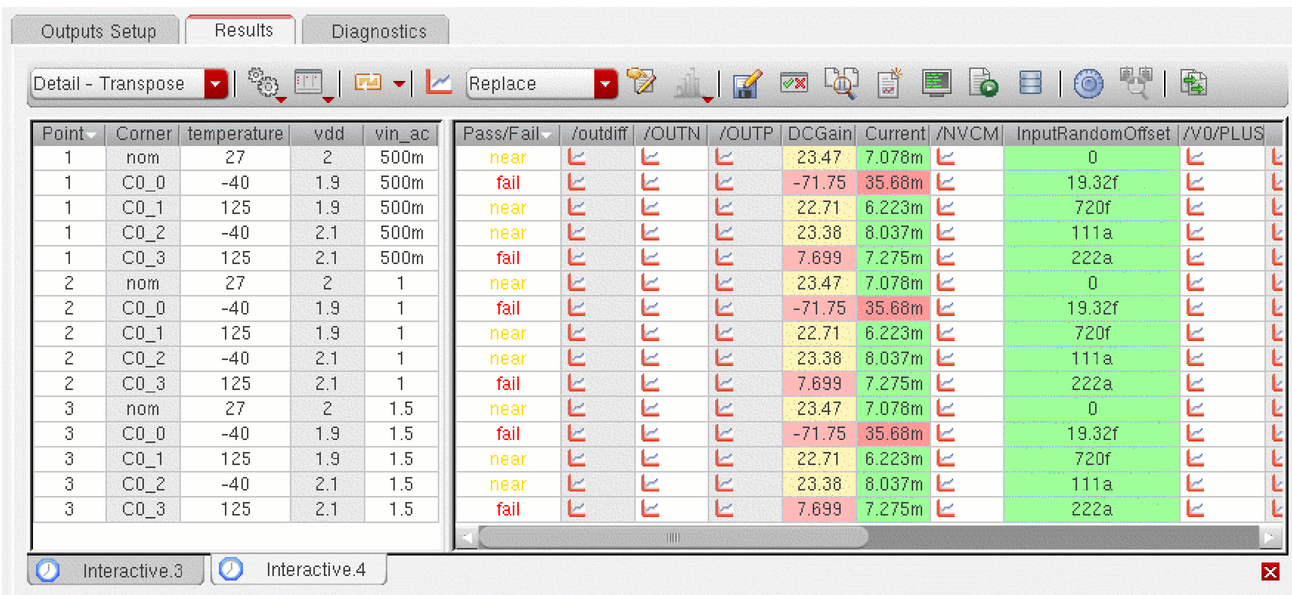


Detail - Transpose

The *Detail - Transpose* view allows visualization of results across corners. The table on the left side displays corner and sweep conditions for each simulation point, and the table on the right side displays the results and *pass*, *near* or *fail* status for output expressions at each simulation point.

Note: The *pass*, *near* or *fail* status displayed for output expressions in the *Pass/Fail* column is based on the status of all specifications across all tests. For more information about the *pass*, *near* or *fail* status for output expressions, see Viewing Specification Results in the Results Tab on page 737.

Figure 14-2 Detail - Transpose View in Results Tab



For more information about the *Detail - Transpose* view, see Using the Detail - Transpose View on page 580.

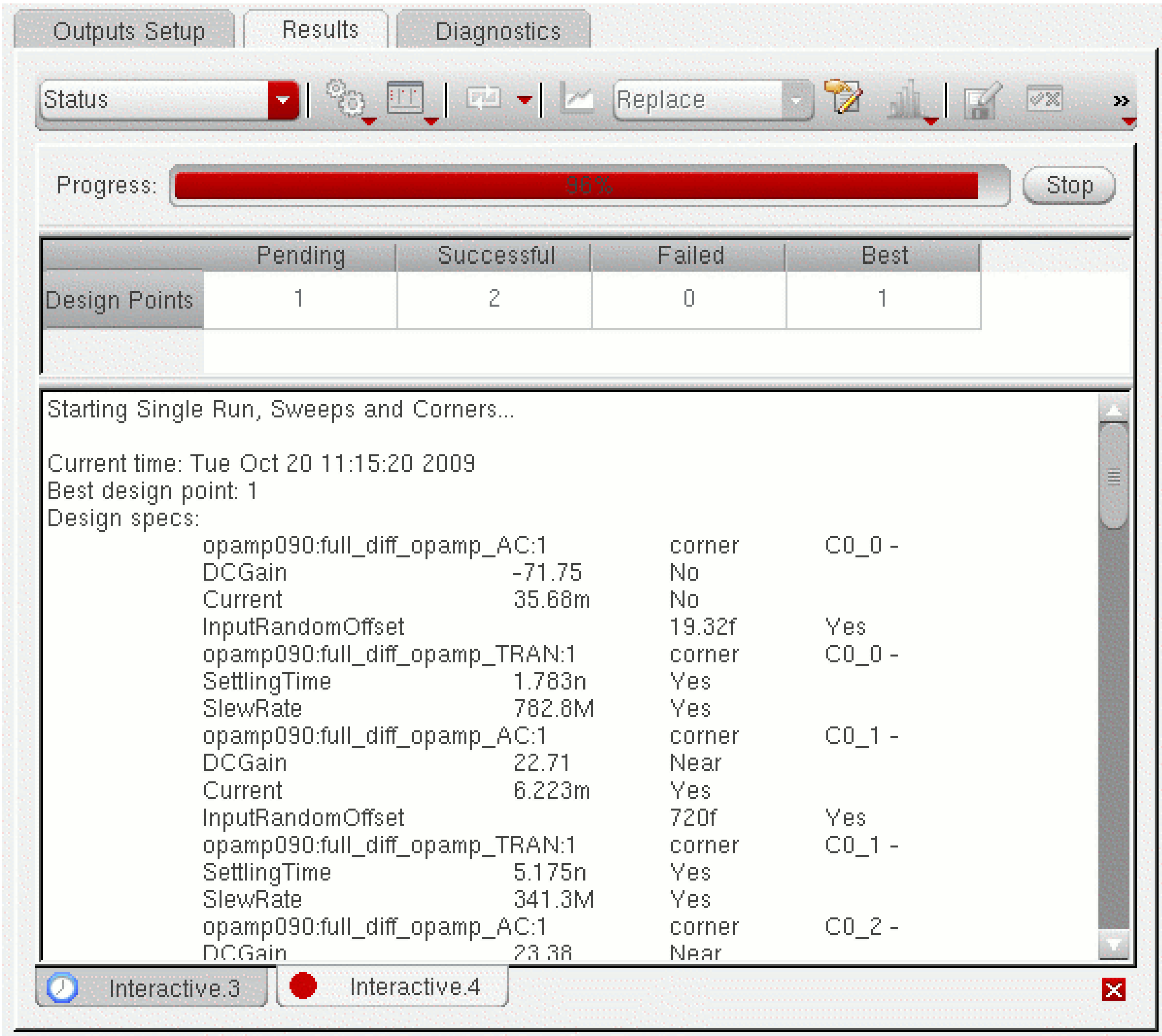
Optimization

The Optimization view displays the *Test*, *Output*, *Value*, *Spec*, *Weight*, *Min*, *Max* and corners-related columns.

Status

The Status view displays the progress and status of the simulation run, and the contents of the run log file. For more information about the Status view, see Using the Status View on page 581.

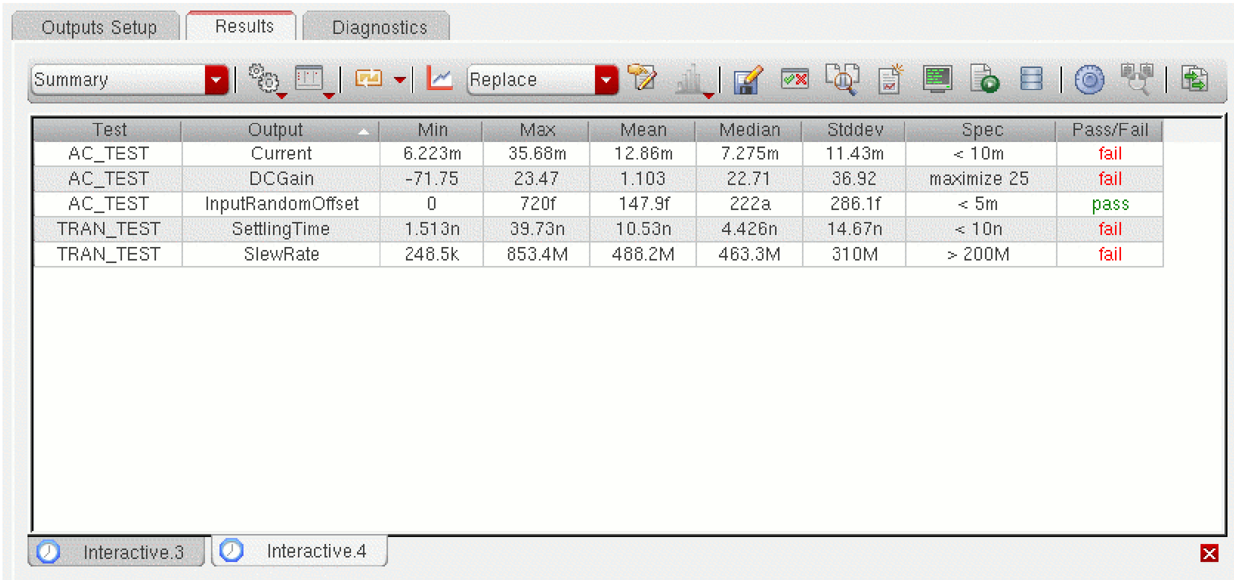
Figure 14-3 Status View in Results Tab



Summary

The Summary view displays the summary of the results for output expressions across all corners and sweeps.

Figure 14-4 Summary View in Results Tab



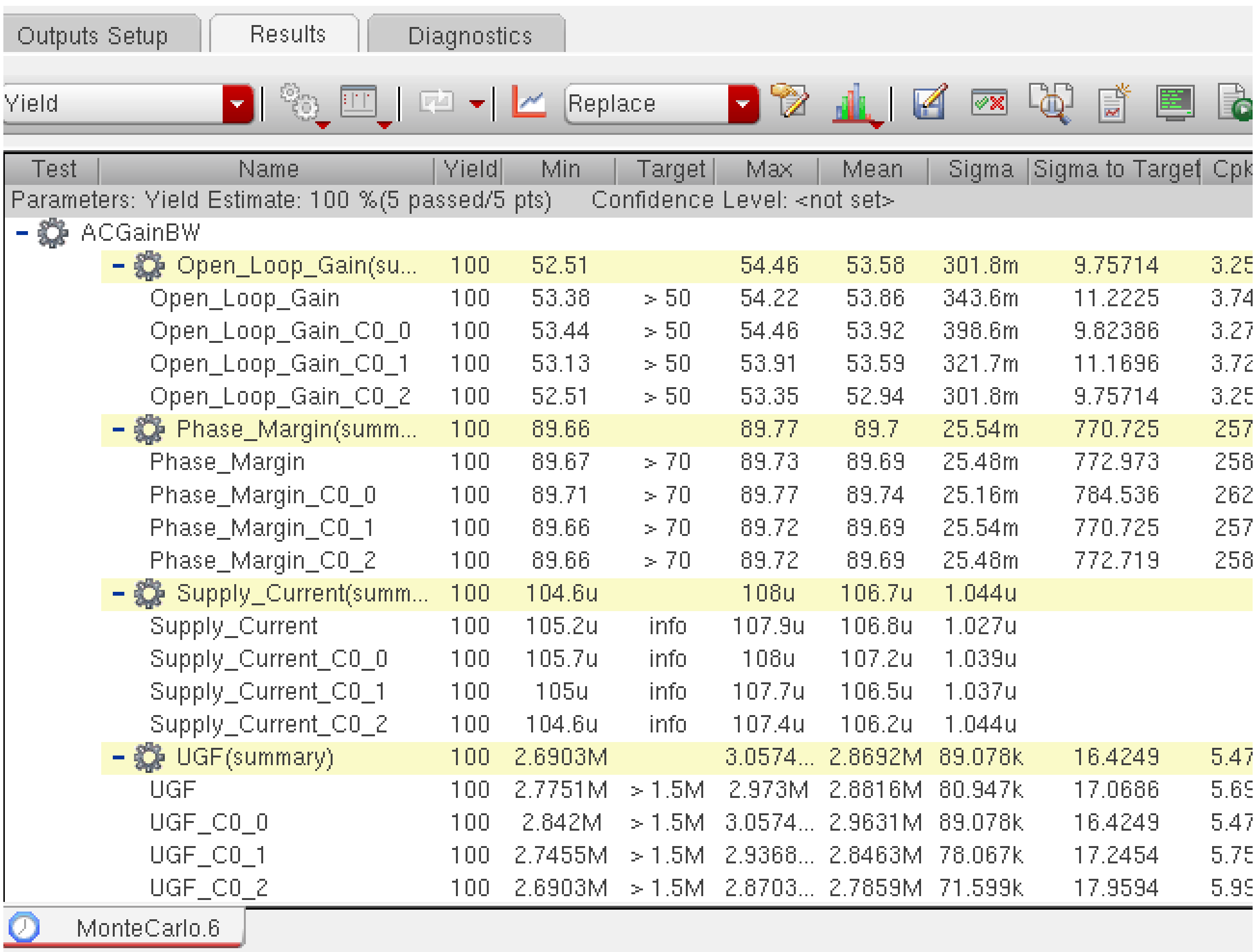
The columns in the Summary view are described below:

Column	Data/Content
Test	Name of the test for which the output is defined.
Output	Name of the output expression.
Min	Minimum value for the specification.
Max	Maximum value for the specification.
Mean	Mean value for the specification.
Median	Median value for the specification.
Stddev	Standard deviation of the values for the specification. **Note:** This is a population standard deviation.
Spec	Specification defined for the output expression.
Pass/Fail	The pass/fail status of the output expression across all sweeps and corners.

Yield

The Monte Carlo Yield view displays for all specifications the overall yield estimate based on pass or fail status. The yield estimate for each specification and statistics, such as mean and standard deviation, are also reported. A sample Yield view is displayed in the following figure.

Figure 14-5 Yield View on Results Tab



Test	Name	Yield	Min	Target	Max	Mean	Sigma	Sigma to Target	Cpk
Parameters: Yield Estimate: 100 %(5 passed/5 pts) Confidence Level: <not set>									
ACGainBW									
	Open_Loop_Gain(su...	100	52.51		54.46	53.58	301.8m	9.75714	3.25
	Open_Loop_Gain	100	53.38	> 50	54.22	53.86	343.6m	11.2225	3.74
	Open_Loop_Gain_C0_0	100	53.44	> 50	54.46	53.92	398.6m	9.82386	3.27
	Open_Loop_Gain_C0_1	100	53.13	> 50	53.91	53.59	321.7m	11.1696	3.72
	Open_Loop_Gain_C0_2	100	52.51	> 50	53.35	52.94	301.8m	9.75714	3.25
	Phase_Margin(summ...	100	89.66		89.77	89.7	25.54m	770.725	257
	Phase_Margin	100	89.67	> 70	89.73	89.69	25.48m	772.973	258
	Phase_Margin_C0_0	100	89.71	> 70	89.77	89.74	25.16m	784.536	262
	Phase_Margin_C0_1	100	89.66	> 70	89.72	89.69	25.54m	770.725	257
	Phase_Margin_C0_2	100	89.66	> 70	89.72	89.69	25.48m	772.719	258
	Supply_Current(summ...	100	104.6u		108u	106.7u	1.044u		
	Supply_Current	100	105.2u	info	107.9u	106.8u	1.027u		
	Supply_Current_C0_0	100	105.7u	info	108u	107.2u	1.039u		
	Supply_Current_C0_1	100	105u	info	107.7u	106.5u	1.037u		
	Supply_Current_C0_2	100	104.6u	info	107.4u	106.2u	1.044u		
	UGF(summary)	100	2.6903M		3.0574...	2.8692M	89.078k	16.4249	5.47
	UGF	100	2.7751M	> 1.5M	2.973M	2.8816M	80.947k	17.0686	5.69
	UGF_C0_0	100	2.842M	> 1.5M	3.0574...	2.9631M	89.078k	16.4249	5.47
	UGF_C0_1	100	2.7455M	> 1.5M	2.9368...	2.8463M	78.067k	17.2454	5.75
	UGF_C0_2	100	2.6903M	> 1.5M	2.8703...	2.7859M	71.599k	17.9594	5.99

Description of the columns displayed in this view is given below:

Column	Data/Content
Test	Test name, top of expandable tree of outputs
Name	Expression/measurement or corner name

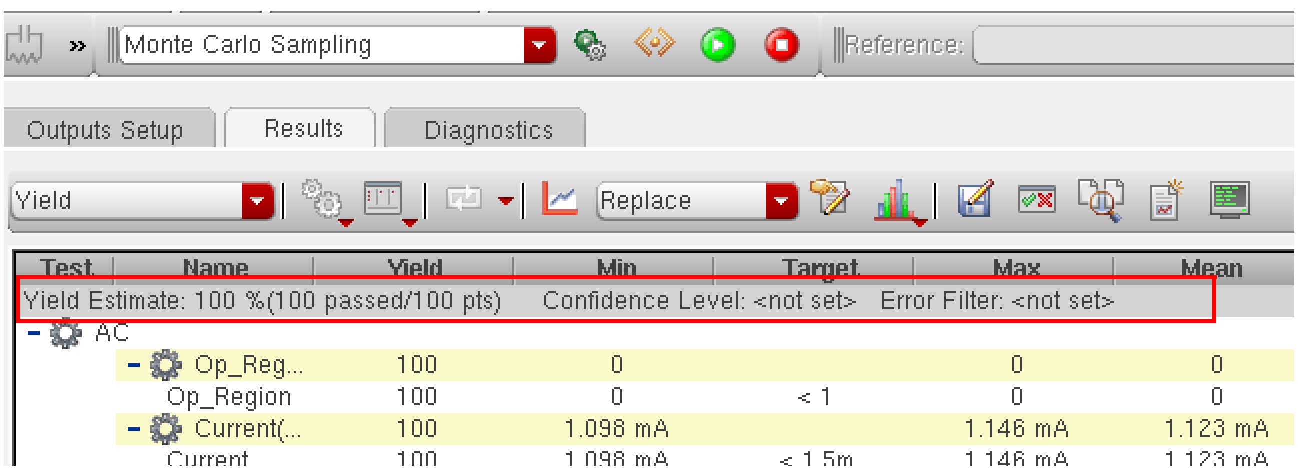
Column	Data/Content
Yield	Resulting yield for each specification
Min	Minimum value resulted for an output
Target	Target specification
Max	Maximum value resulted for an output
Mean	Mean value
Sigma	Standard deviation
Sigma to Target	Distance from the sigma value to the target value, defined as follows: `mean value to target value / standard deviation`
Cpk	Cpk is the process capability index, a statistical measure of the process capability. It is the capability of a process to produce an output within the given specification limits. If USL and LSL are the upper and lower specification limits, μ is the estimated mean of the process and σ is the estimated variability of the process, Cpk is calculated by using the following equation. $$Cpk = \min\left[\frac{USL-\mu}{3\sigma}, \frac{\mu-LSL}{3\sigma}\right]$$ Cpk is a good estimation of yield if the output distribution is Gaussian. `Cpk > 1` indicates greater than 3-sigma yield and `Cpk < 0` indicates that the process mean falls outside the USL or LSL limits.

The summary row for each specification displays the analysis for that specification across all corners. Summary values are derived as follows:

Column	Summary value
Yield	Minimum yield value across all corners
Min	Minimum value across all corners
Max	Maximum value across all corners
Mean	Mean value across all corners

Column	Summary value
Sigma to Target	Minimum sigma to target value across all corners **Note:** This is a sample standard deviation.
Sigma	Sigma value corresponding to the minimum sigma to target value across all corners **Note:** This is a sample standard deviation.
Cpk	Minimum Cpk value across all corners

The gray-colored row at the top displays the *Yield Estimate* – an estimate of the circuit yield taking into account all of the specifications.



By default, the results displayed on this tab do not consider a confidence level. You can specify a confidence level for which you want to view the results. ADE XL displays a range of mean and sigma values for which the specifications are met. For more details, refer to Viewing Data for a Specific Confidence Interval.

You can also set the error filter to filter out the results with simulation error or evaluation errors. For more details, refer to Filtering Out Error Data from the Yield View.

Tip

To specify the names of columns you want to view in the Yield view, on the Results toolbar, click *Configure what is shown in the table* and select the names of columns to show. Alternatively, you can specify the names of columns to show by setting the yieldViewShowDefault environment variable.

Using the Detail - Transpose View

Most of the tasks that you can perform in the *Detail - Transpose* view are the same as in other results views. The following topics describe the tasks that are specific to the *Detail - Transpose* view, or different from those in other results views:

- Hiding and Showing Columns in the Detail - Transpose View on page 580
- Changing the Order of Columns in the Detail - Transpose View on page 580
- Viewing the Test Name for Outputs in the Detail - Transpose View on page 580

Hiding and Showing Columns in the Detail - Transpose View

To hide a column, do the following:

➤ Right-click on the name of the column and choose *Hide Column*.

To display all the columns, do the following:

➤ Right-click on the name of any column and choose *Show All Column*s.

Changing the Order of Columns in the Detail - Transpose View

To change the order of columns, do the following:

1. Click on a column name.
2. Drag and drop the column at the desired location.

Viewing the Test Name for Outputs in the Detail - Transpose View

To view the name of the test for which an output is defined, do the following:

➡ Place the mouse pointer on the name of the column for the output on the right hand side of the Detail - Transpose view.

The name of the test for which the output is defined is displayed in a pop-up.

Important

In the Detail - Transpose view, you can also view the statistical parameter values used on each point and create a statistical corner that can be used to rerun a particular point in the distribution. These commands are available in the right-click pop-up menu commands.

Using the Status View

See the following topics for more information about using the *Status* view on the Results tab of the Outputs pane:

- Stopping the Simulation Run from the Status View on page 581
- Hiding and Showing Information in the Status View on page 581

Stopping the Simulation Run from the Status View

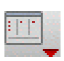
To stop the simulation run, do the following:

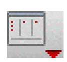
➡ Click the *Stop* button next to the *Progress* bar.

Hiding and Showing Information in the Status View

Click the  button on the Results tab. A check mark next to an item indicates that information for that item is being displayed.

To hide information, do the following:

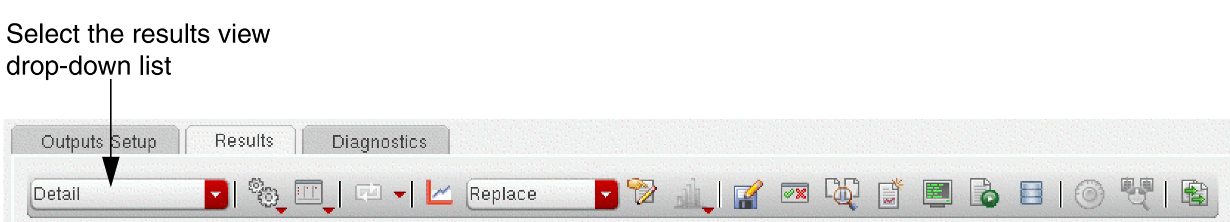
➤ Click the  button on the Results tab and choose the item.

To unhide the information click the  button once again and choose the item.

Switching Between Results Views

To switch between views, do the following:

➤ From the *Select the results view* drop-down list, select the view you want to display.



Freezing Columns in the Detail and Optimization View

You can view two areas of the Results tab and lock the columns in one area by freezing columns. When you freeze columns, you select specific columns that remain visible when scrolling in the Results tab.

Note: You can freeze columns only in the Detail and Optimization views in the Results tab.

To freeze columns, do the following:

➡ Right-click on the column to the right of where you want the split to appear and choose *Freeze columns*.

For example, to freeze the *Output* column and all the columns to the left of the *Output* column, right-click on the heading for the *Output* column and choose *Freeze columns*. This allows you to view the contents of the *Output* column and all the columns to the left of the *Output* column when you use the horizontal scroll bar in the Results tab scroll through the results for corners.

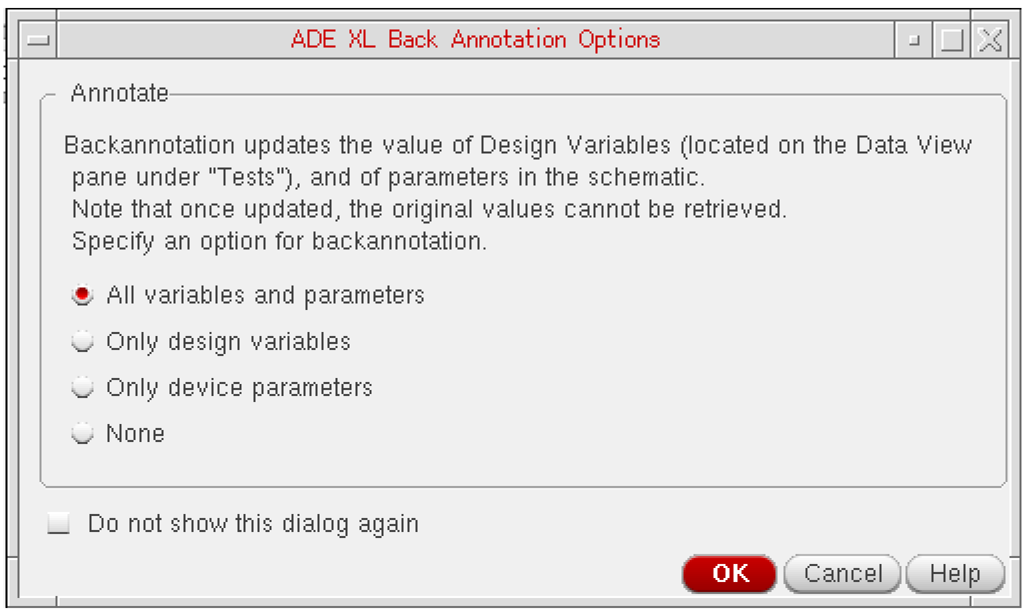
Backannotating from ADE XL Results

After running an ADE XL simulation, you can backannotate the results from the best or the desired design point to make appropriate changes in the design and simulation setup. When you backannotate the results from a design point, the values of global variables used for that point are copied to the *Design Variables* list in the Data View pane of the ADE XL setup and the values of the device parameters are copied to the schematic view of the design.

To backannotate the ADE XL results, do the following:

1. Ensure that the Details view is open in the Results view tab.
2. Right-click on the gray bar on top of the selected design point and choose *Backannotate*.

   The ADE XL Back Annotation Options form is displayed, as shown below.



3. Select an appropriate option on this form to specify the types of values that you want to backannotate from the results:

   ❑ *All variables and parameters*: This option backannotates the values of all the global variables and device parameters. This option is selected by default.

   Note: You can change the default value by using the `defaultBackAnnotationOption` environment variable.

   ❑ *Only design variables*: This option backannotates only the global variables.

- ❑ *Only device parameters*: This option backannotates only the device parameters.
- ❑ *None*: This option does not backannotate any value.

4. Select the *Do not show this dialog again* check box if you do not want to view this form every time you backannotate values.

   The check box will be set for the session only. Once you close ADE XL and open the next session, you will have to set the check box again.

5. Click *OK* to apply the changes and close the form.

Note: You cannot restore the original values after the results are backannotated to the schematic or the ADE XL setup.

Selecting the Plot Mode

On the Results tab of the Outputs pane, do the following:

➤ From the *Select the plotting mode* drop-down list on the Results tab, select one of the following plot modes:

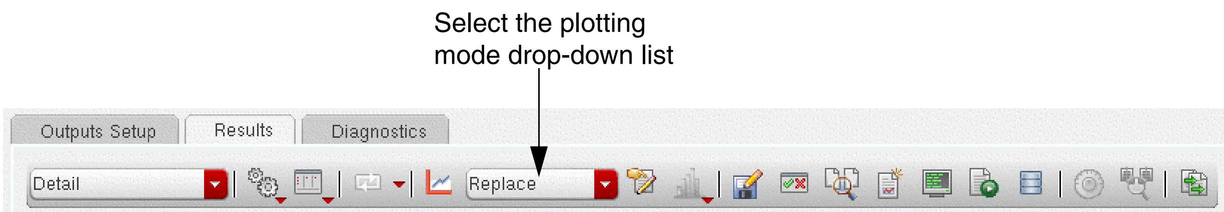


Table 14-2 Plot Modes

Plot Mode	Description
Append	A new trace is appended to the active waveform window if the test name and X-axis match. ***Important Points To Note*** ■ Traces for different tests are always plotted in different waveform windows tabs in the Virtuoso Visualization and Analysis XL window. ■ If the test name for the active waveform window tab is different from the test name of the new trace to be plotted, ADE XL looks for an existing tab (from left) for the same test. If found, the new trace is appended to that window tab. Otherwise, a new window tab is created. ■ The axes units of the new trace must match with the axes of the existing traces. If the units of X-axis differ, the new trace is plotted in a new subwindow in the existing window tab. If the units of Y-axis differ, a new Y-axis is created in the existing subwindow.
Replace	New traces replace the existing traces in the active waveform window. If the active waveform window has multiple subwindows, all the subwindows are deleted and the new traces are plotted.

Plot Mode	Description
New SubWin	New traces are plotted in a new subwindow in the active waveform window
New Win	New traces are plotted in a new waveform window

Note: These plot modes can also be set from the Direct Plot, Calculator, and Results Browser windows.

Setting Default Plotting Options for All Tests

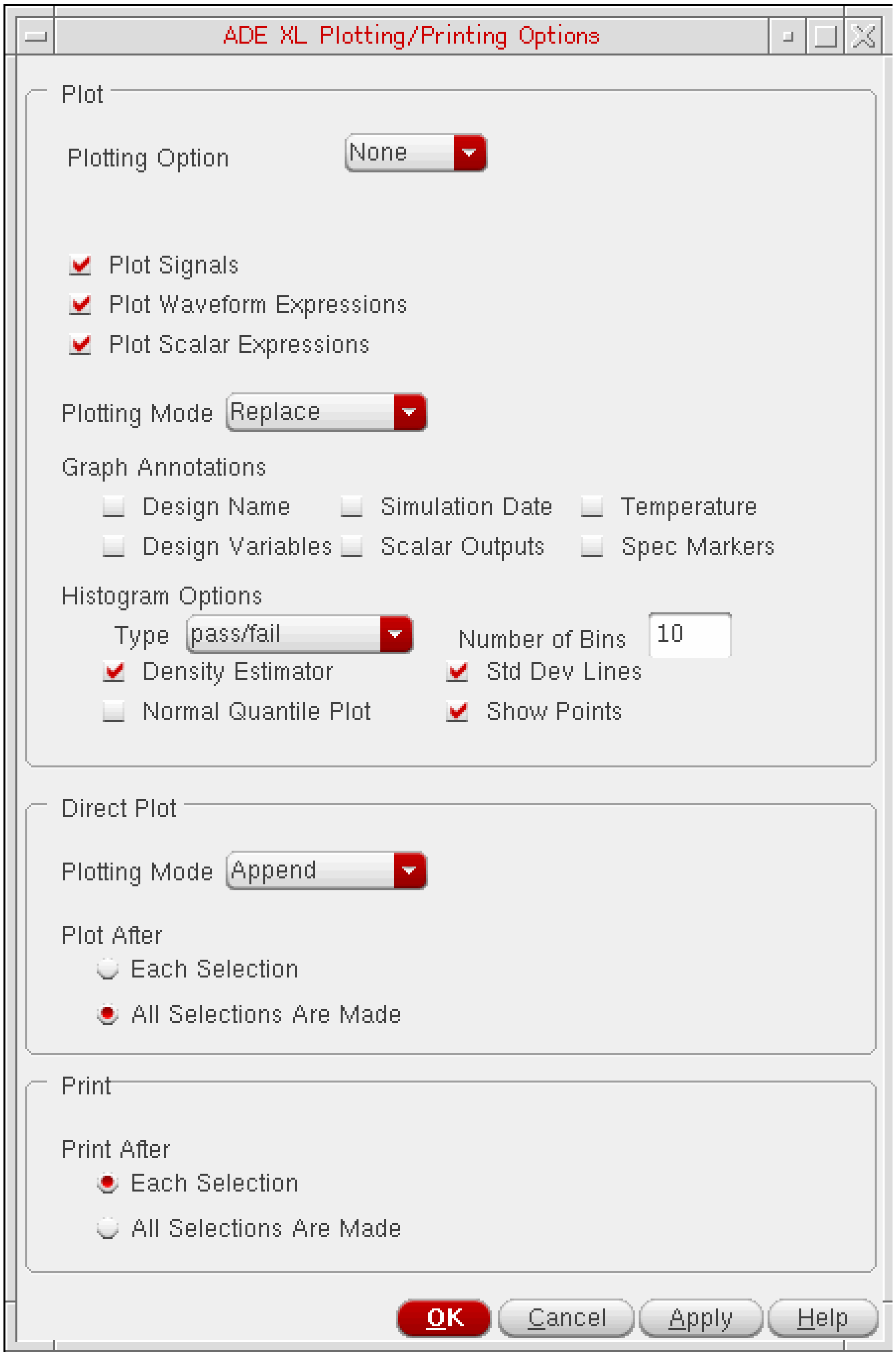
You can use the ADE XL Plotting/Printing Options form to set the default printing and plotting options for all tests. These options will be used:

- For automatic plotting of results after a simulation run is complete
- When you click the *Plot all waveforms* button on the Results tab

Note: You can use the Setting Plotting Options form to override the default printing and plotting options for individual tests. For more information, see Setting Plotting Options for Specific Tests on page 620.

1. To open the ADE XL Plotting/Printing Options form, do one of the following:
   - Choose *Options – Plotting/Printing*.
   - Click the button on the Outputs Setup tab or the Results tab of the Outputs pane.

The ADE XL Plotting/Printing Options form appears.



2. Specify plotting options.

See the following topics for assistance:

- ❑ Specifying the Default Plotting Option for All Tests on page 590

- ❑ Specifying the Outputs that Need to be Plotted on page 591
- ❑ Specifying the Default Plotting Mode for All Tests on page 592
- ❑ Specifying Annotations for the Graph Window on page 592
- ❑ Specifying the Default Direct Plot Mode on page 593
- ❑ Specifying When to Plot Direct Plot Results on page 594
- ❑ Specifying When to Print Results on page 594

3. When you are finished specifying options, click *OK*.

Specifying the Default Plotting Option for All Tests

You can specify the default plotting option for all tests. These options will be used for the following:

- Automatic plotting of results after a simulation run is complete
- When you click the *Plot all waveforms* button on the Results tab

To specify the default plotting option for all tests, do the following in the ADE XL Printing/Plotting Options form:

➡ In the *Plot* group box, select the plotting option from the *Plotting Option* drop-down list.

Select	To
None	Disable automatic plotting of results after the simulation run. This is the default option.
Auto	Automatically plot outputs after the simulation run. For every subsequent simulation run, a new graph replaces the existing graph. You can choose to append the new graph to the existing graph of the previous simulation run or plot it in a new window using the *Plotting Mode* drop-down list. When this option is selected, any customization done in the Virtuoso Visualization and Analysis graph windows that are currently open, such as setting up traces, colors, or zoom levels, are not reused. Every time a graph is plotted, the default settings are used.

Select	To
Refresh	Plot the results by updating the existing graphs in the Virtuoso Visualization and Analysis XL graph window that is already open. Use this option when you want to save and reapply graph and trace settings on the plots that you want to review across different simulation runs. **Note:** Currently, this feature is supported only in the `Single Run, Sweeps, and Corners` run mode. For more details on how the graphs are refreshed, refer to Refreshing Graphs on page 594. This mode ignores the following options in the ADE XL Printing/Plotting Options form: ■ Plotting mode selected in the *Plotting Mode* drop-down list ■ *Plot Signals*, *Plot Waveform Expressions* and *Plot Scalar Expressions* options ■ *Graph annotation* options

You can configure the default plotting option by using the `plotType` environment variable.

Return to main procedure.

Specifying the Outputs that Need to be Plotted

To specify the outputs that need to be plotted automatically after the simulation finishes, do the following in the *Plot* group box:

Select	To
Plot Signals	To plot signals.
Plot Waveform Expressions	To plot expressions that evaluate to waveforms.
Plot Scalar Expressions	To plot expressions that evaluate to scalar values.

Return to main procedure.

Specifying the Default Plotting Mode for All Tests

To specify the default plotting mode for all tests, do the following:

➡ In the *Plot* group box, select the plot mode from the *Plotting Mode* drop-down list.

For more information about plot modes, see Table 14-2 on page 585.

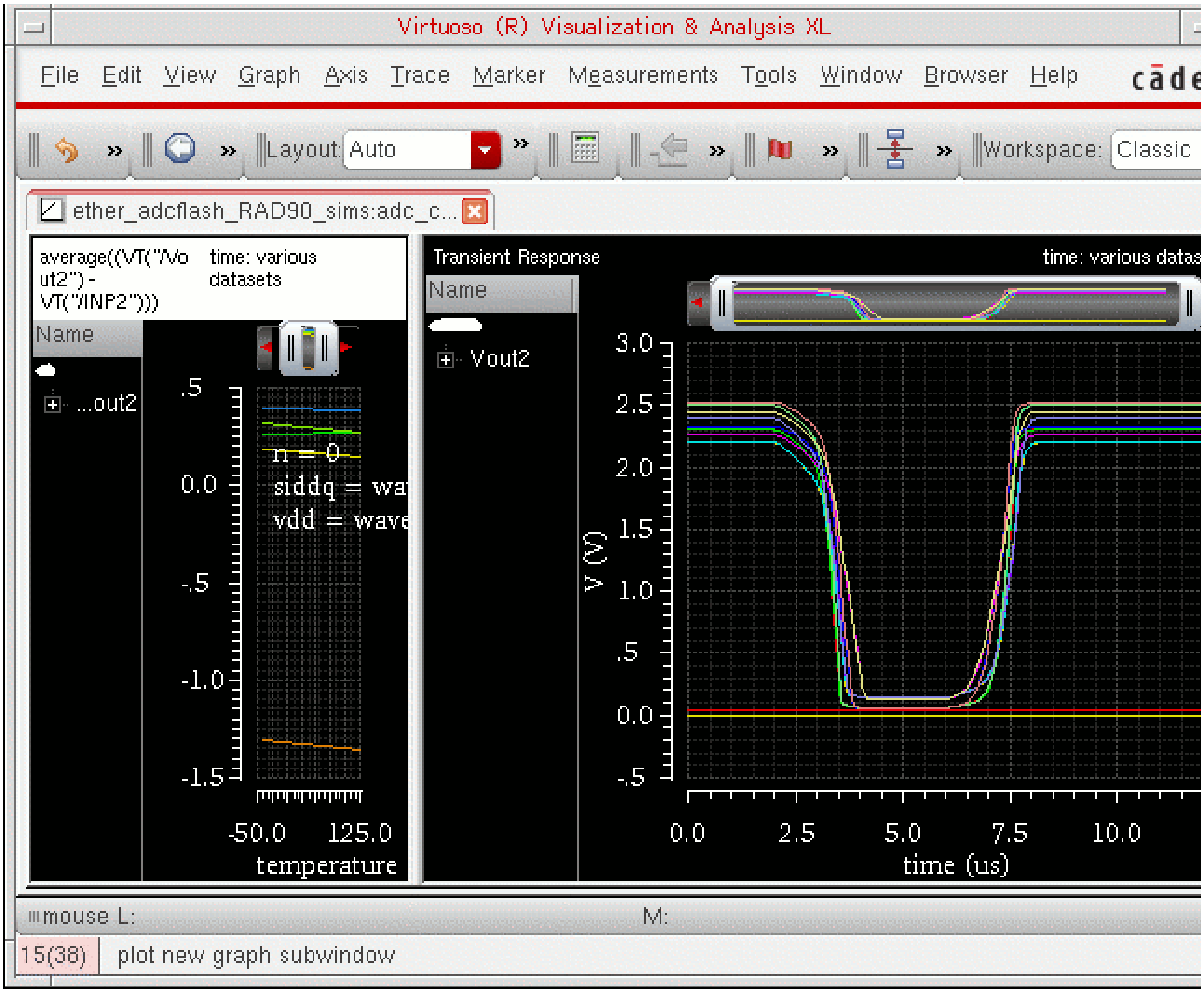
Return to main procedure.

Specifying Annotations for the Graph Window

To specify annotations for the graph window, do the following:

1. Select one or more of the following *Annotations* check boxes:
   - ❑ *Design Name* — Displays design name in the title banner of the waveform window.
   - ❑ *Simulation Date* — Displays date and time of simulation in the title banner of the waveform window.
   - ❑ *Temperature* — Displays simulation temperature in the plot area of the waveform window.
   - ❑ *Design Variables* — Displays design variables and their values in the plot area of the waveform window.
   - ❑ *Scalar Outputs* — Displays calculated results in the plot area of the waveform window.
   - ❑ Spec Markers — Displays spec markers in the plot area of the waveform window. For more details about the spec markers, refer to Displaying Spec Markers on Graphs on page 609.
2. Click *Apply*.

Here is a waveform window with all annotations applied:



Return to main procedure.

Specifying the Default Direct Plot Mode

To specify the default direct plot mode for all tests, do the following:

➡ In the *Direct Plot* group box, select the plot mode from the *Plotting Mode* drop-down list.

For more information about plot modes, see Table 14-2 on page 585.

Return to main procedure.

Specifying When to Plot Direct Plot Results

To specify when to plot results you select from your schematic when you use the *Results – Direct Plot* submenu, do the following:

1. Select one of the following *Direct Plots Done After* options:
   - ❑ *Each Selection* – Plot results after each item you select on the schematic.
   - ❑ *All Selections Are Made* – Plot results after you select all items and press *Esc* to end selection mode.
2. Click *Apply*.

Return to main procedure.

Specifying When to Print Results

To specify when to print results to the Results Display Window, do the following:

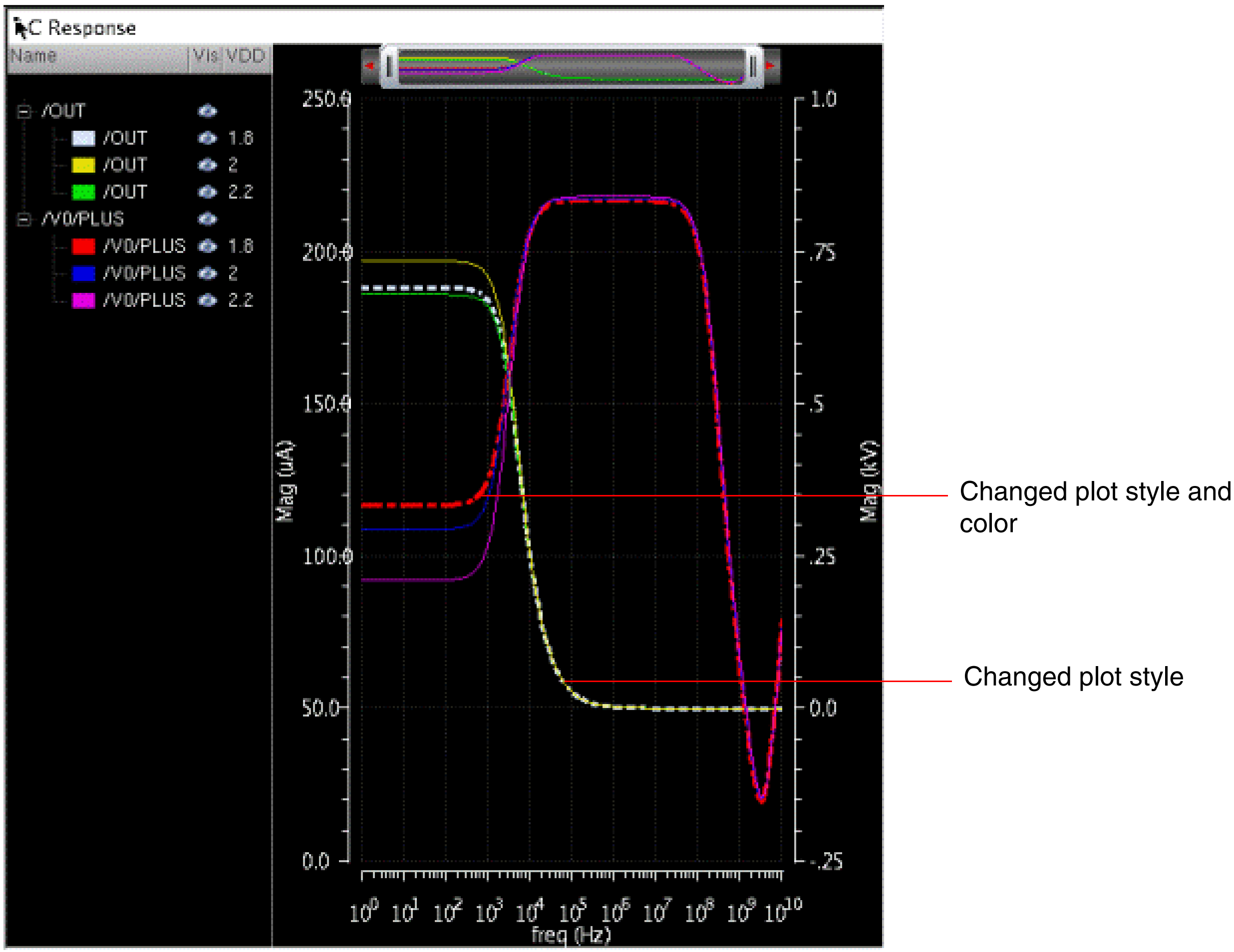
1. Select one of the following *Print After* options:
   - ❑ *Each Selection* – Print results after each item you select on the schematic.
   - ❑ *All Selections Are Made* – Print results after you select all items and press *Esc* to end selection mode.
2. Click *Apply*.

Return to main procedure.

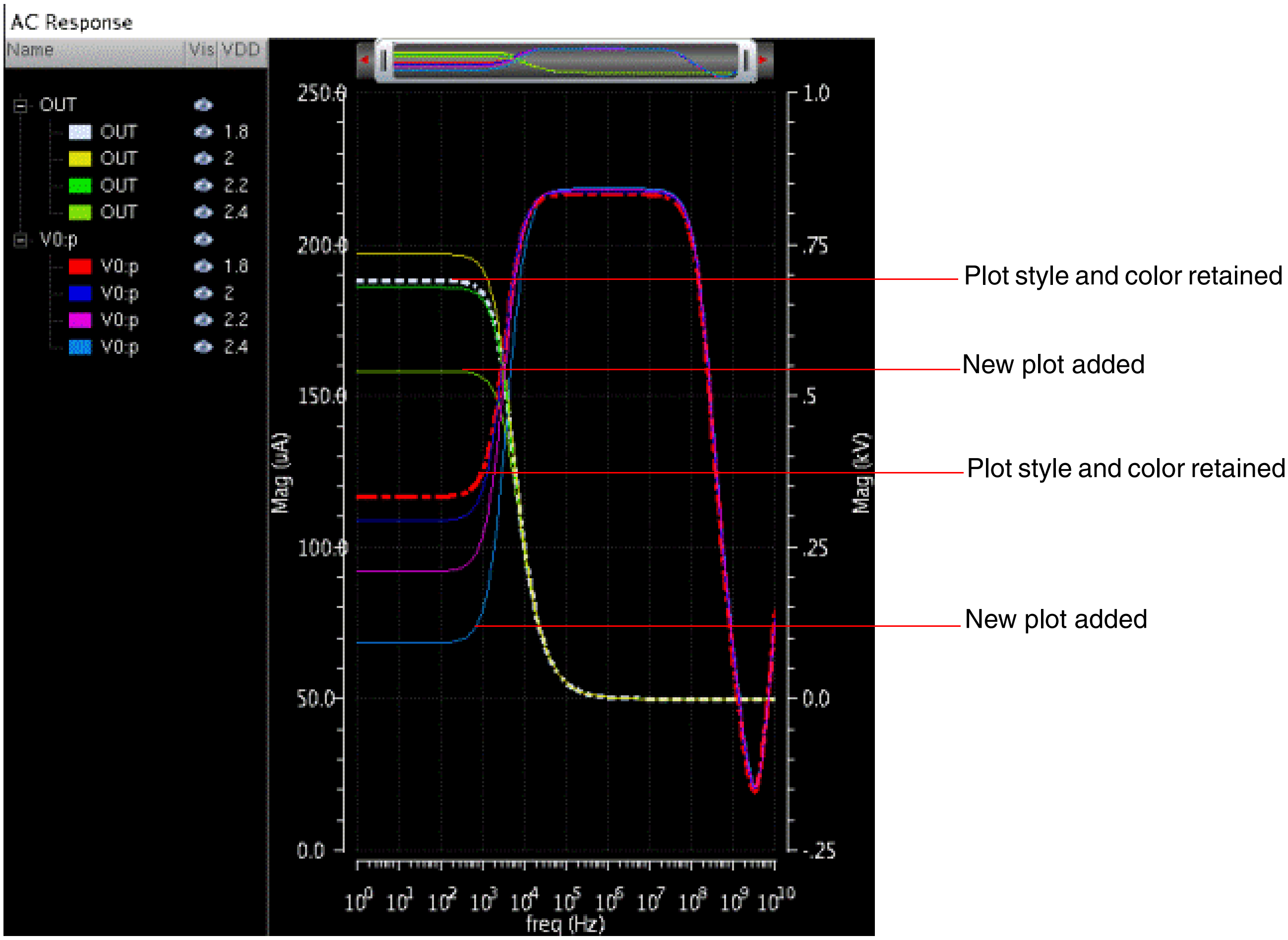
Refreshing Graphs

Graphs can be refreshed by using the `Refresh` plotting option in the ADE XL Printing/Plotting Options form. This option updates already open graphs with new simulation data and retains graph and trace settings.

You can use the `Refresh` plotting option to review graphs across different simulation runs. For example, for a test `ACGainBW`, you want to run multiple simulations with varying values of the global variable *`VDD`*. In the first run, you sweep *`VDD`* for three values 1.8, 2.0, and 2.2. After the simulation results are plotted in the graph, you customize two plots, as shown in the figure below.


C Response
Name
Vis
VDD
/OUT
/OUT 1.8
/OUT 2
/OUT 2.2
/V0/PLUS
/V0/PLUS 1.8
/V0/PLUS 2
/V0/PLUS 2.2
Mag (uA)
Mag (kV)
freq (Hz)
Changed plot style and color
Changed plot style

Now, do not close this graph and set *Refresh* in the *Plotting Option* field. In the next run, sweep *`VDD`* for four values: `1.8`, `2.0`, `2.2`, and `2.4`. After simulation, the graph appears as shown in the following figure.



Note that:

- Traces for *`VDD`* = `1.8`, `2.0`, and `2.2` are updated in the same graph and trace settings are retained.
- New trace for *`VDD`* = `2.4` is added to the same graph

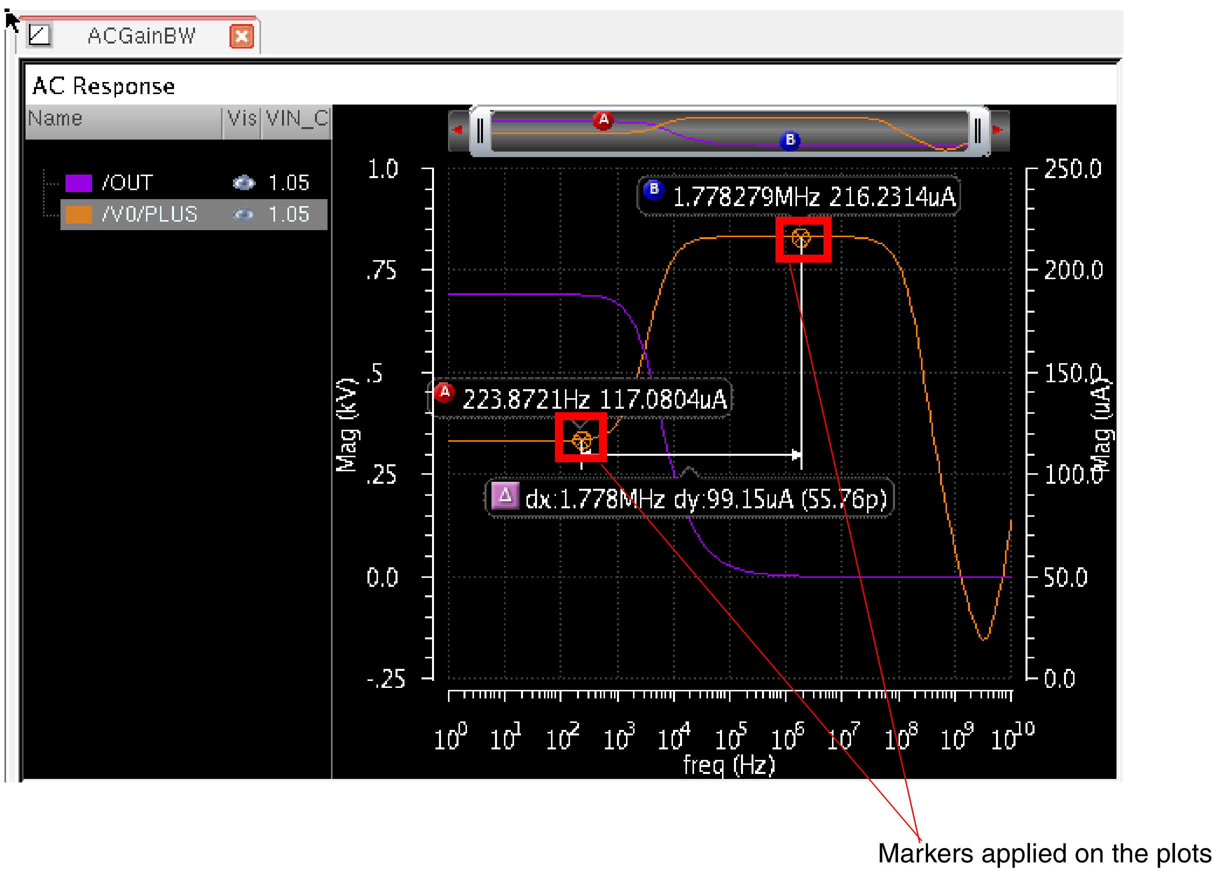
Graph Settings Retained During Refresh Plotting

When the *Plotting Option* field is set to `Refresh`, the graphs retain the following settings across simulation runs:

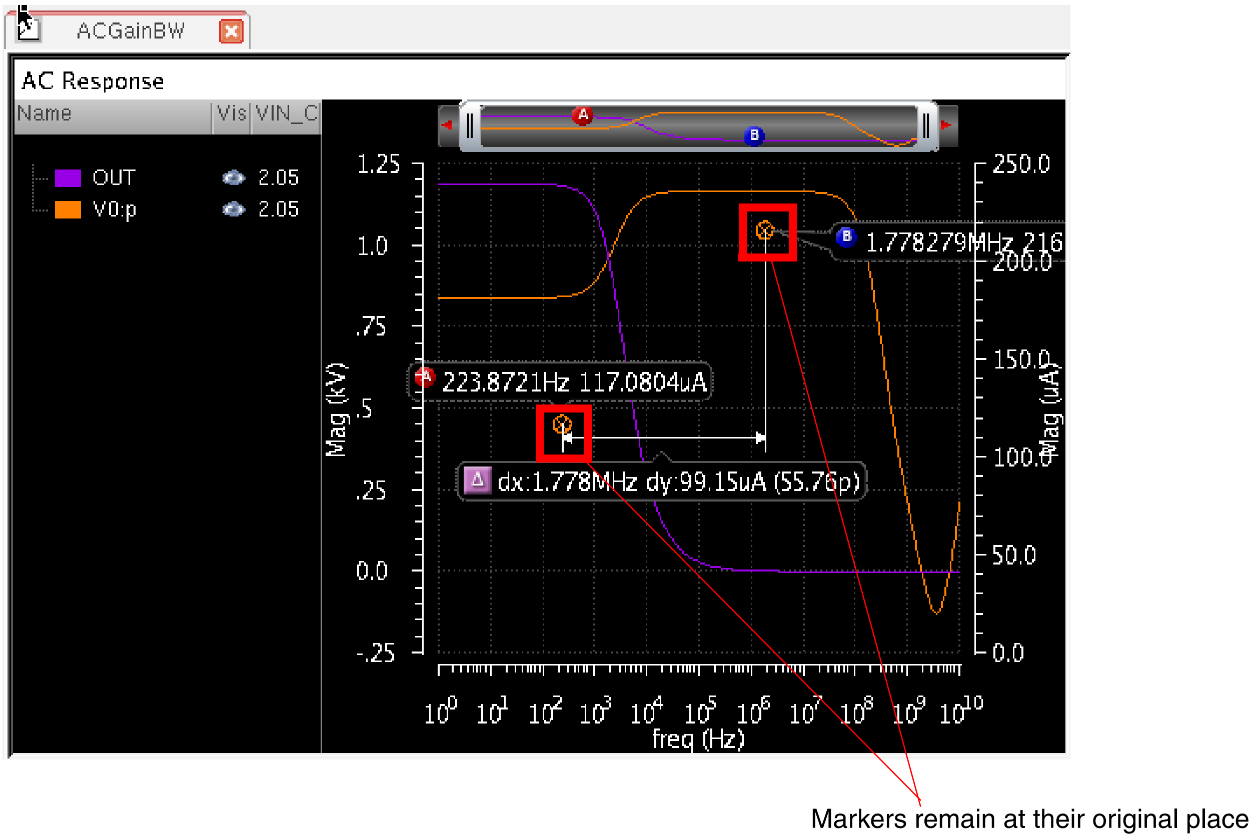
- Trace color, type, style, width, or symbols

- Visibility status of graphs
- Axes settings
- Pan and zoom settings
- Graph layout
- Strip layout
- Markers and marker locations

The following example shows how markers are retained with refresh plotting. Add an AB delta marker on the `V0/PLUS` trace in your graph, as shown in the following figure.



Now, if you change the values of variables and run the simulation again, the refreshed graph is plotted for the new simulation results, as shown in the figure below.



Note that the A and B markers remain displayed at their original positions and are not connected with the trace in the refreshed plot.

Important

If you save graphs as part of an ADE XL state, all graph settings specified above are retained and when you reload the state, the trace settings are retrained. However, if you save the graph (in a .grf file) from standalone graph window, these settings are not saved.

Graph Settings Not Retained During Refresh Plotting

The following graph settings are not retained:

- Swapping of sweep variable on the X-axis
- Addition of a new graph window

- Any plot added to the graph in any of the following ways:
    - By using the Direct Plot main form
    - By using the Plot or Plot Across Corners commands on the Results tab
    - By plotting directly from the Results Browser
    - By plotting from the Calculator. For example, by plotting analog to digital, digital to analog, eye diagram, or spectrum measurements.
    - By adding outputs from a different history

    These graphs are not refreshed because any plot that is not on the Outputs tab of ADE XL is deleted after the graph is refreshed. To retain these additional plots and their settings, you can send the plot from the graph to the ADE XL *Outputs* tab by using the *Send to ADE* command on the shortcut menu of the graph. For more details, refer to Sending Traces to ADE in *Virtuoso Visualization and Analysis XL User Guide*.

The following sections describe how the graphs are updated with refresh plotting in different scenarios:

- Refresh Plotting with Varying Tests
- Refresh Plotting with Varying Analyses
- Refresh Plotting with Varying Output Setup
- Refresh Plotting with Varying Sweep Variables
- Refresh Plotting with Varying Corners

Refresh Plotting with Varying Tests

The following table summarizes how graphs are refreshed when tests are varied across different simulation runs:

Table 14-3 Effect of Test Variations on Graph Settings

Test Variation	Effect on Graph Settings
Add or enable a test	■ A new graph window is added to display the plots of the newly added/enabled test. ■ Graph windows for the already existing tests are retained. Their graphs are updated with the new simulation data and trace settings are retained.

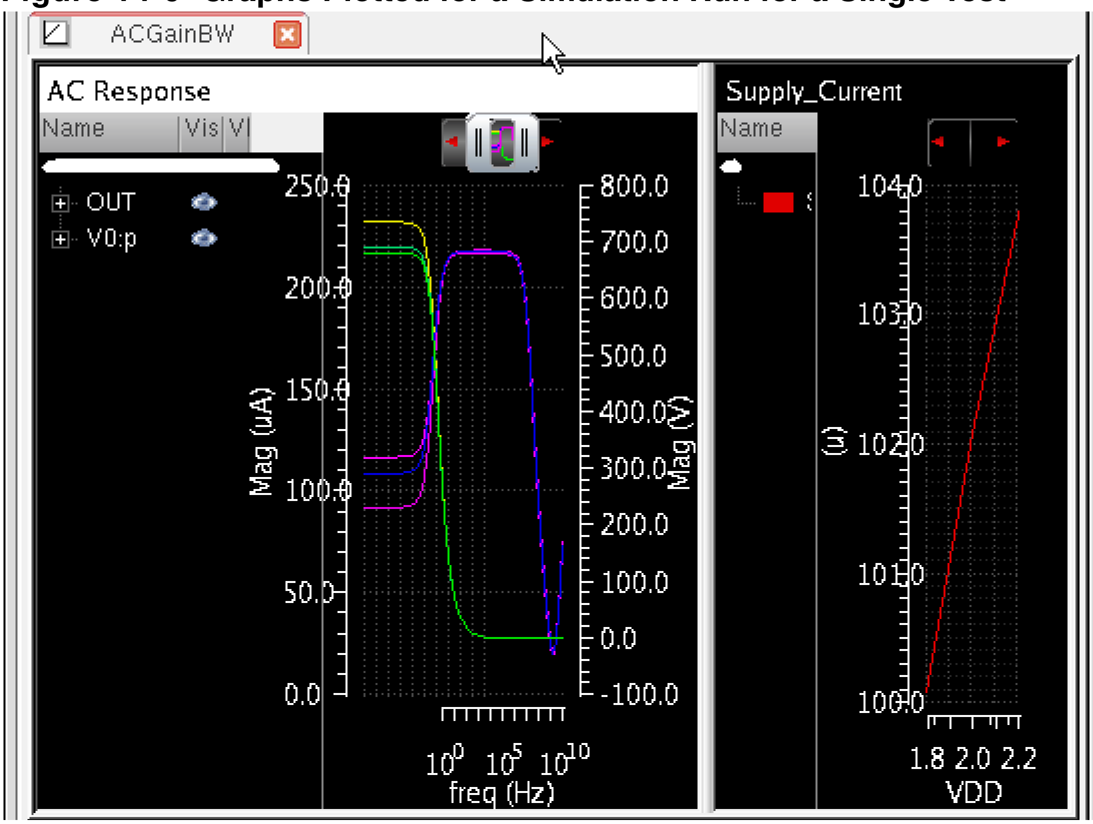
Table 14-3 Effect of Test Variations on Graph Settings

Test Variation	Effect on Graph Settings
Change the test for which simulation is run	■ A new graph window is added to display the plots of the new/changed test. ■ Graph windows displayed in the previous run are retained.
Delete or disable a test	■ Graph window for the test that is deleted or disabled is retained, but their graphs are not updated.

The following example shows how graphs are updated when a new test is added:

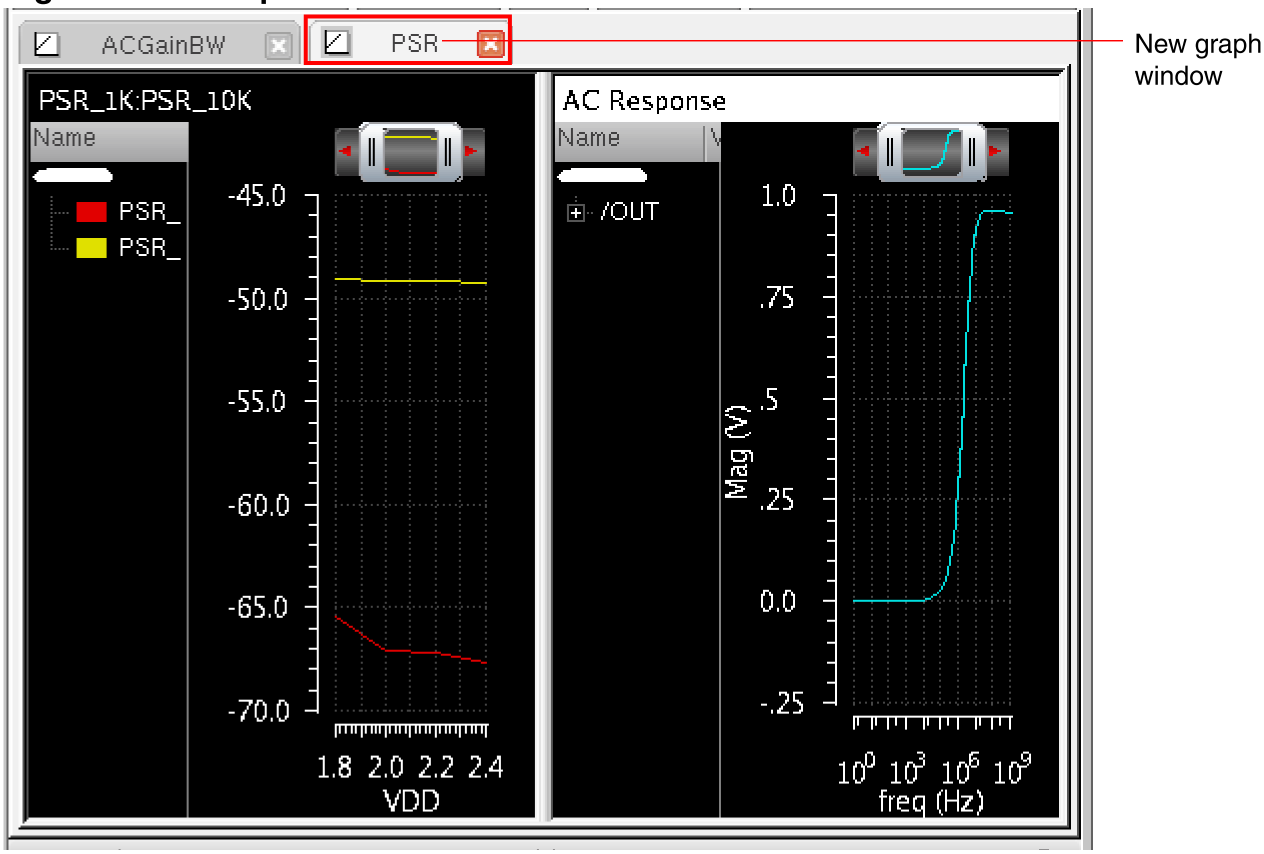
When you run simulation with a single test, the graph appears as shown in the following figures.

Figure 14-6 Graphs Plotted for a Simulation Run for a Single Test



Now, if you add one more test, `PSR`, and run simulation, a new graph window is added, as shown in the following figure.

Figure 14-7 Graphs Plotted for Simulation Run for an Additional Test



Refresh Plotting with Varying Analyses

The following table summarizes how graphs are refreshed when analyses are varied across different simulation runs (assuming that the test remains same):

Table 14-4 Effect of Analyses Variations on Graph Settings

Analyses Variation	Effect on Graph Settings
Add or enable an analysis	■ A new subwindow is added to the graph to display the plots of the newly added or enabled analysis. ■ Graphs for the analysis that are common across the two runs are updated with the new simulation data and any trace settings are retained.
Delete or disable an analysis	■ Subwindows related to the analysis that has been deleted or disabled after the previous simulation run are removed from the graph.

Refresh Plotting with Varying Output Setup

The following table summarizes how graphs are refreshed when outputs are varied across different simulation runs:

Table 14-5 Effect of Changes in Output Variations on Graph Settings

Output Variation	Effect on Graph Settings
Add or enable an output	■ Newly added or enabled output is added in a new subwindow. ■ Existing graphs are updated with new simulation data and their graph and trace settings are retained.
Delete or disable an output	■ The plot for the deleted or disabled output is removed. ■ Other graphs are updated with new simulation data and their graph and trace settings are retained.

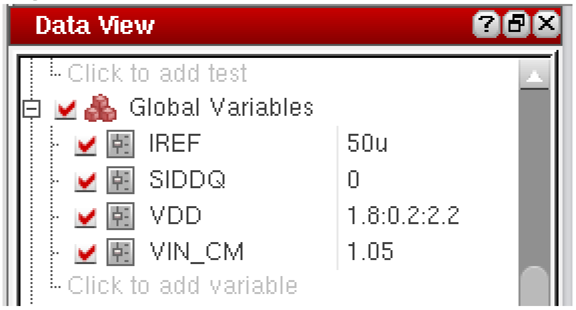
Note: The Refresh plotting option does not update the graphs plotted for MATLAB measurements.

Refresh Plotting with Varying Sweep Variables

If you vary sweeps across simulation runs, the traces that use common sweep values are updated with new simulation data. For new sweep values, new traces are added to the same graph. Traces for unmatched sweep values are removed.

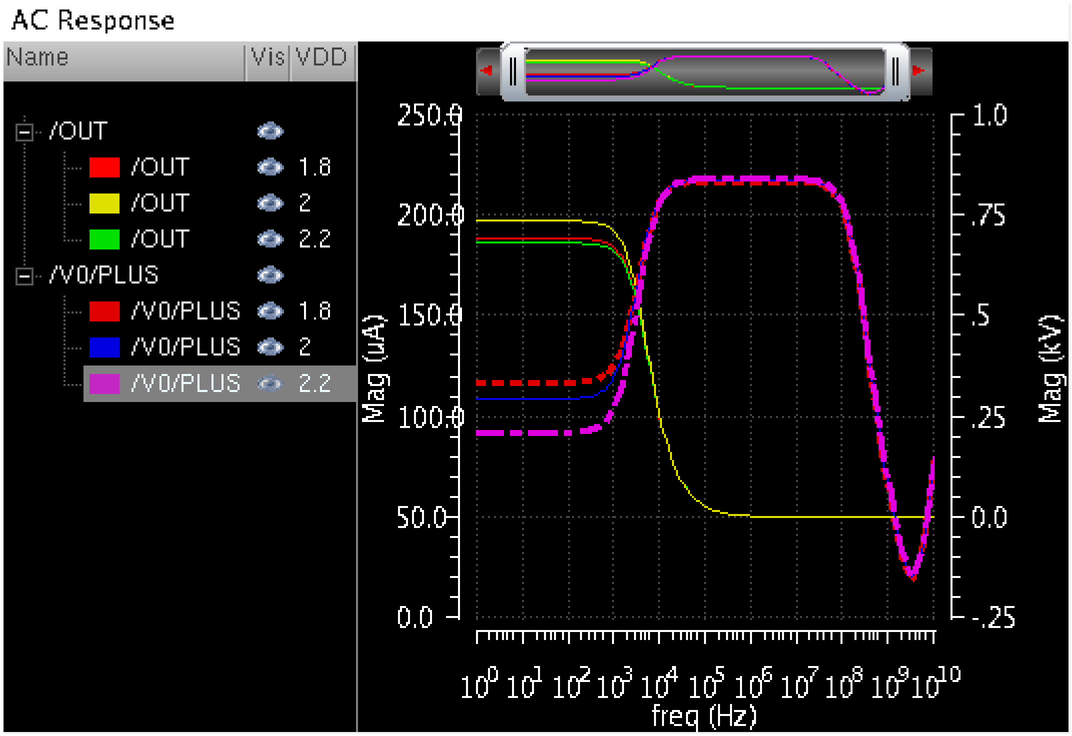
For example, set the variables as shown below.

Figure 14-8 Variables Used for Simulation



The graph plotted in this case includes plots for all three sweep values for *VDD* as shown below.

Figure 14-9 Graphs Generated after Sweeping Variable VDD



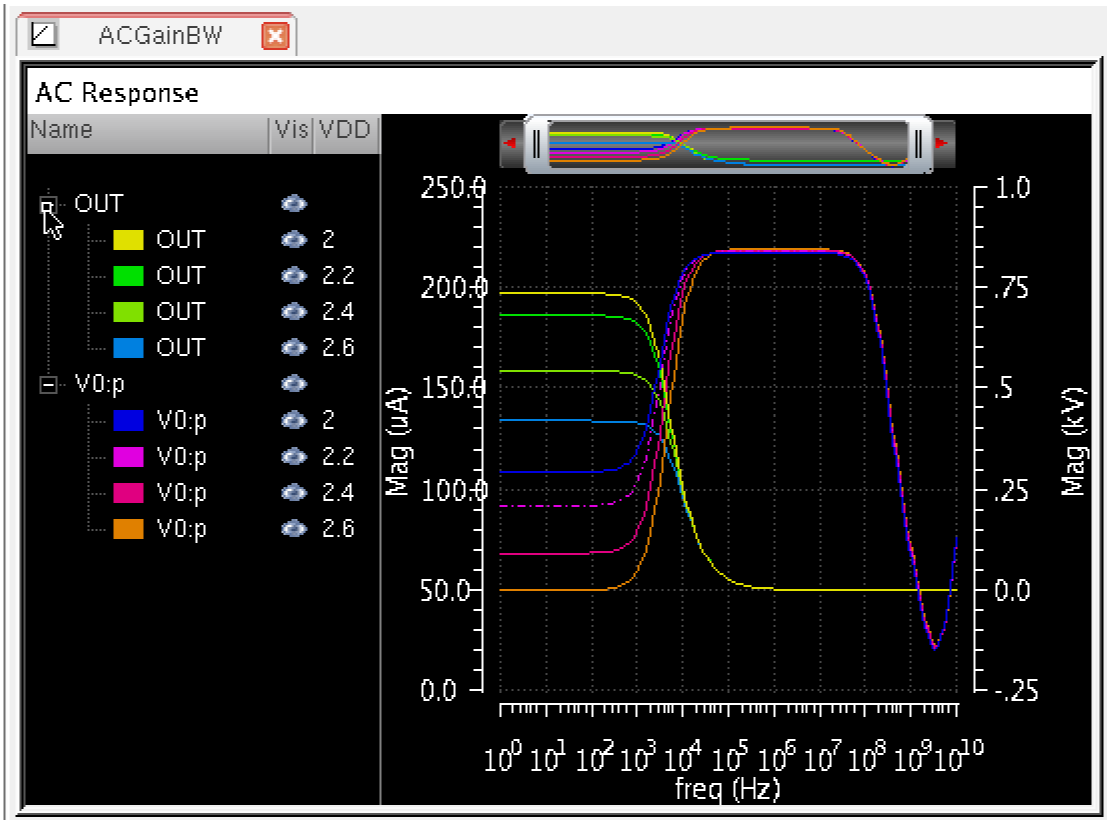
For the next simulation run, change the sweep values for *VDD* to `2.0:0.2:2.6`. Note the change in values of *VDD*, as shown below (mismatch values are underlined).

Values of *VDD* in the first run: <u>`1.8`</u>, `2.0`, and `2.2`

Values of *VDD* in the second run: `2.0`, `2.2`, <u>`2.4`</u>, and <u>`2.6`</u>

With the Refresh plotting option, the traces generated for the matching values of *VDD*, that is, when *VDD* is `2.0` and `2.2`, are updated with new simulation data. The trace that was earlier plotted for *VDD*=1.8 is deleted. New traces are added for *VDD*= `2.4` and `2.6`.

Figure 14-10 Graph settings Saved for the Plots with Common Sweep Values



However, there are some exceptions to this. In some cases, the traces plotted for common sweep values of the swept variables are not refreshed. These cases are listed below:

- Change from a single run to the sweep run and vice-versa.
- Change in the number of swept variables. For example, in the first simulation run, you sweep variable x, and in the subsequent runs, you sweep variables x and y. In this case, traces plotted for x are not refreshed.

Refresh Plotting with Varying Corners

Note: The following scenarios are based on the assumption that the test name remains same because only the traces for same test are plotted in the same waveform window. Traces from different tests are plotted in different waveform windows.

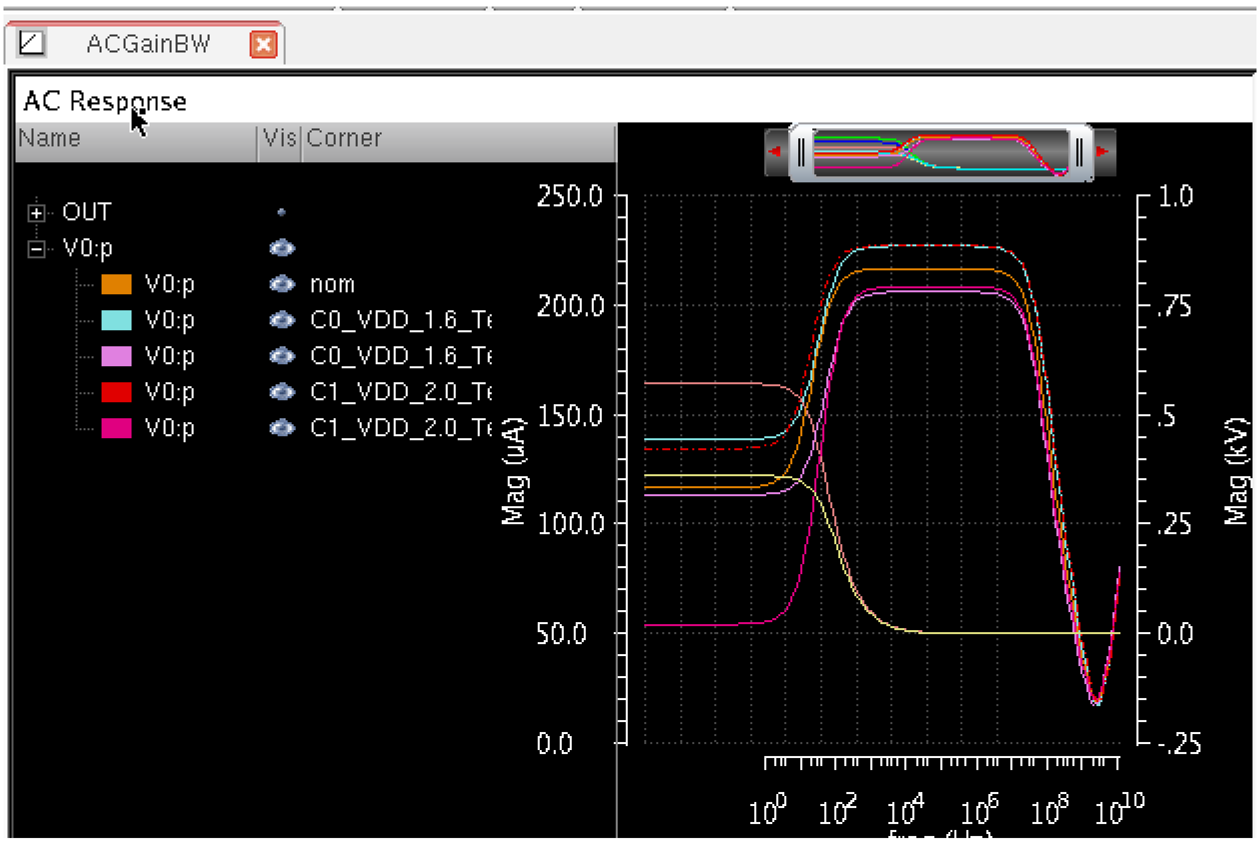
If you vary corners across simulation runs, traces generated for the common set of corners are updated with the new simulation data. For other corners, new traces are added in the same graph. Traces for unmatched corners are deleted.

The following example shows are graphs are updated with different corners across three different runs:

Run 1:

Run a simulation for three corners: nominal, `C0`, and `C1`. Both `C0` and `C1` sweep temperature for two values, `-25` and `75`, and use a common section from the model file, but vary the value of *VDD*. The graph is plotted as shown below.

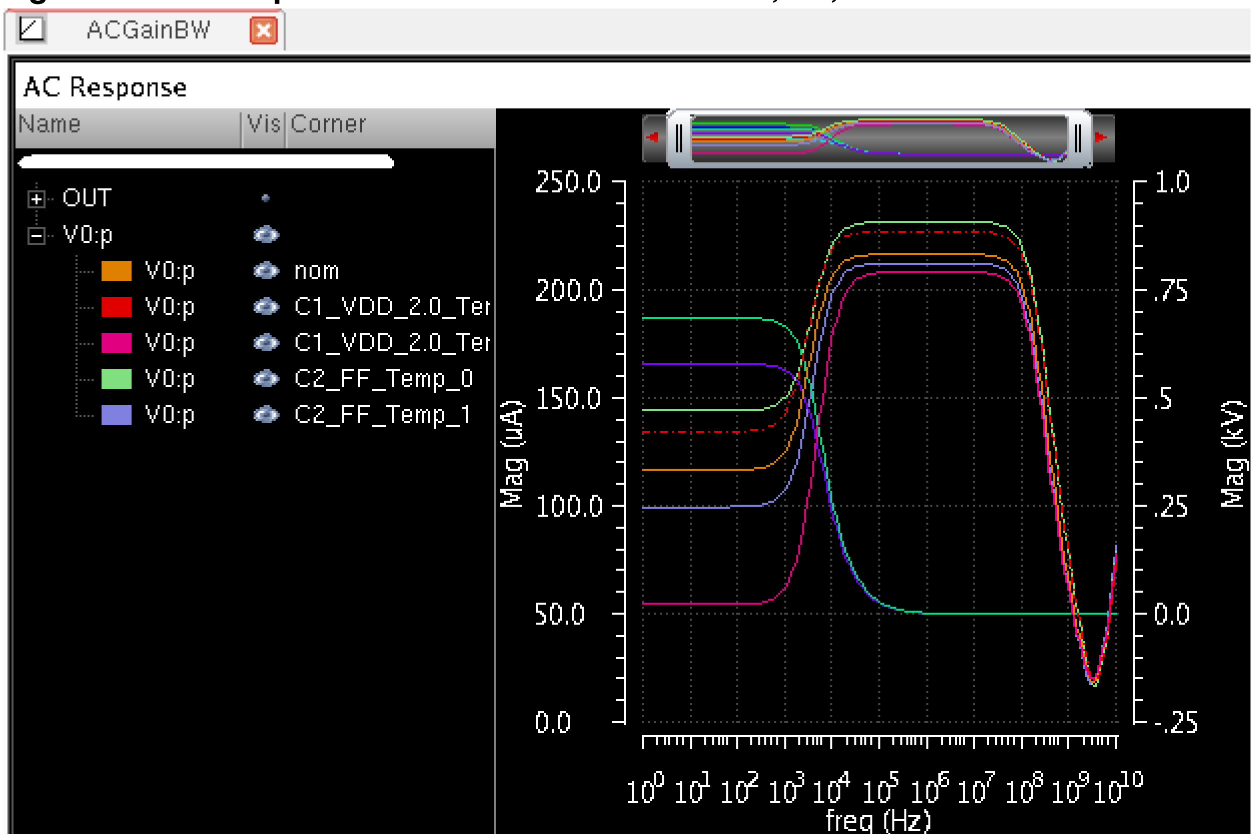
Figure 14-11 Graphs Plotted for Corners: nominal, C0, and C1



Run 2:

For the next simulation run, use the `nominal` and `C1` corners. In place of corner `C0`, use corner `C2`, which uses the same value of *VDD* as is being used in `C1`, but a different section from the model file. After simulation, the graph is updated as shown below.

Figure 14-12 Graphs Plotted for Corners: nominal, C1, and C2



Note the following changes:

- Traces plotted for the corners, `nominal` and `C1` have been retained.
- New traces have been plotted for corner `C2`.
- Traces for corner `C0` have been removed from the graph.

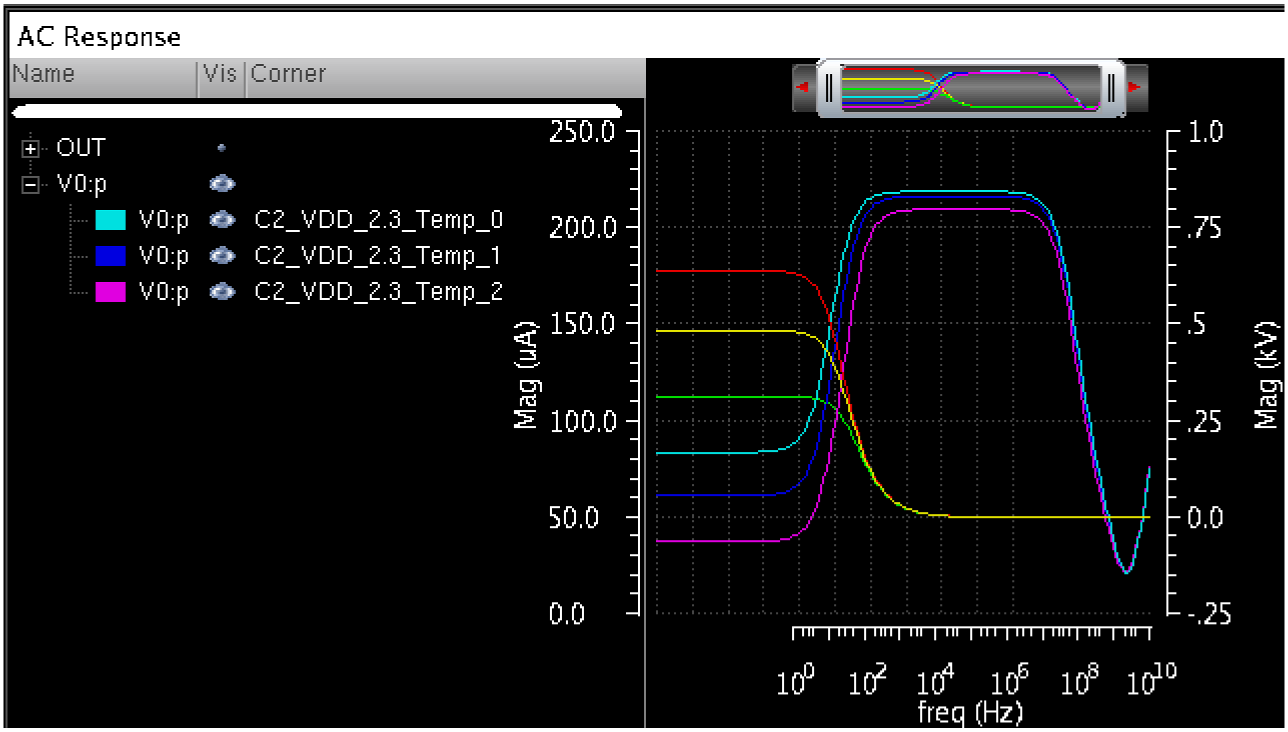
Run 3:

For the third run, use change the values of corner variables for corner `C2` and disable all other corners. Use the following variables for corner `C2`:

`VDD` = `2.3`
`Temp` = `25, 40, 75`

In this case, corner conditions of the new simulation run do not match with that of the previous run. Therefore, no trace is updated and only new traces are plotted, as shown below.

Figure 14-13 Graphs Plotted For Corner C2



The following table summarizes how graphs are updated when corners and sweep variables are varied across different simulation runs:

Table 14-6 Effect of Corners and Sweep Variations on Graph Settings

Corner or Sweep Variations Across Two Simulation Runs	Effect on Graph Settings
In the first run, a single run is used with the nominal corner. In the next run, the global variable is swept and the same nominal corner is used	■ Traces for the sweep values that are same across both the runs are refreshed and their trace settings are retained. For example, in the first run, `VDD=1.8`, and in the second run, `VDD=1.8`, `2.0`, and `2.2`. In this case, trace plotted for `VDD=1.8` is refreshed. ■ If there is no common value for the swept variable across different run, new traces are plotted and settings are not retained. For example, in the first run, `VDD=1.8`, and in the second run, `VDD=2.0`, `2.2`, and `2.4`. In this case, trace plotted for `VDD=1.8` is removed.

Table 14-6 Effect of Corners and Sweep Variations on Graph Settings

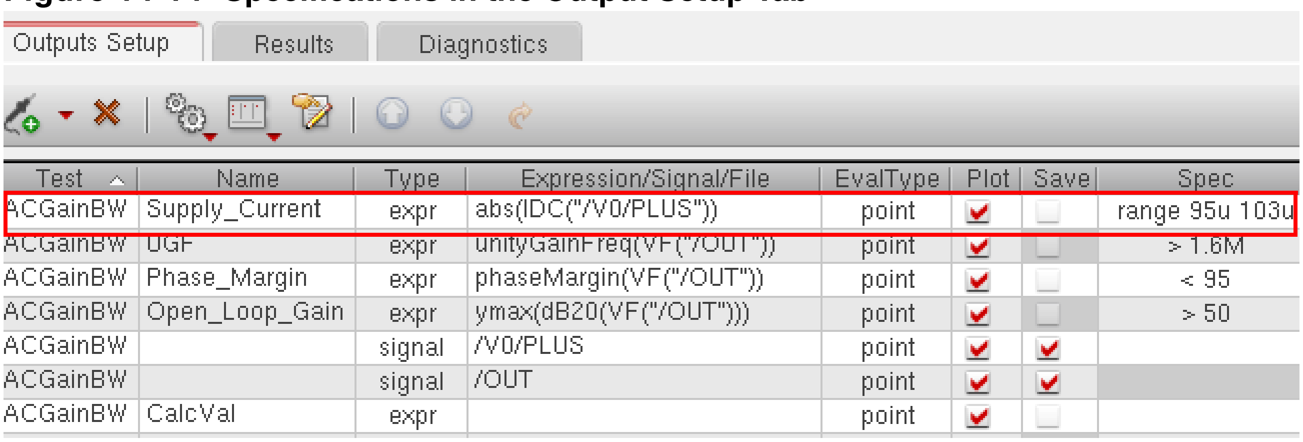
Corner or Sweep Variations Across Two Simulation Runs	Effect on Graph Settings
Both simulation runs sweep a common global variable. Nominal corner is same in both the runs.	■ Traces for the sweep values that are same across both the runs are refreshed and their trace settings are retained. ■ If none of the sweep values match across the runs, new traces are plotted and no settings are not retained.
In the first run, a corner is used. In the next run, a new corner is added to the existing setup.	■ Traces plotted for the existing corner are updated and their settings are retained. ■ For the new corner, new traces are plotted in the same graph.
Corners used in the first run are disabled and a new corner is added.	■ No traces are updated with new simulation data. ■ New traces are plotted in the same graph.
In both simulation runs, only nominal corner is used, but the values of variables in the nominal corner are different.	■ No traces are updated with new simulation data. ■ New traces are plotted in the same graph.
Both simulation runs use the same set of sweep values for global variables and nominal corner. In addition, the second run uses an additional corner.	■ Traces plotted for the nominal corner are updated with new simulation data and their settings are retained. ■ For the new corner, new traces are plotted in the same graph.

Displaying Spec Markers on Graphs

Spec markers are color markers that are displayed on a graph to demarcate the region under a trace as pass or fail region. This demarcation is done by shading the pass and fail regions with green and red line patterns, respectively. The demarcation depends on the status of simulation results with respect to the specifications given on the Outputs Setup tab.

For example, in the Outputs Setup tab, a specification is set for `Supply_Current` in the *Spec* column, as shown in the figure below.

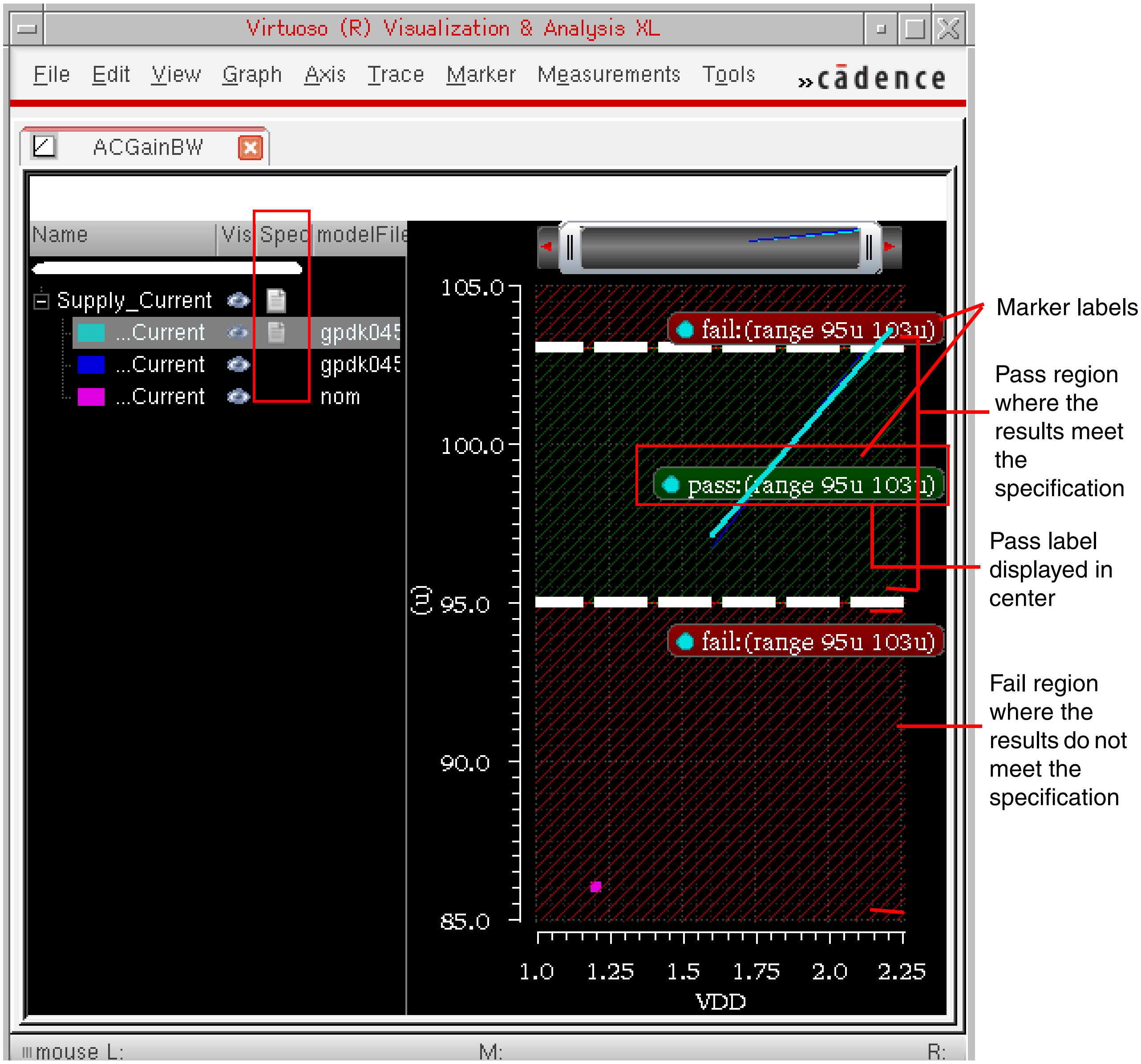
Figure 14-14 Specifications in the Output Setup Tab



Test	Name	Type	Expression/Signal/File	EvalType	Plot	Save	Spec
ACGainBW	Supply_Current	expr	abs(IDC("/V0/PLUS"))	point	✓		range 95u 103u
ACGainBW	UGF	expr	unityGainFreq(VF("/OUT"))	point	✓		> 1.6M
ACGainBW	Phase_Margin	expr	phaseMargin(VF("/OUT"))	point	✓		< 95
ACGainBW	Open_Loop_Gain	expr	ymax(dB20(VF("/OUT")))	point	✓		> 50
ACGainBW		signal	/V0/PLUS	point	✓	✓	
ACGainBW		signal	/OUT	point	✓	✓	
ACGainBW	CalcVal	expr		point	✓		

When you plot the results in Virtuoso Visualization and Analysis XL window, spec markers are displayed for `Supply_Current`, as shown in the figure below.

Figure 14-15 Spec Markers on a Graph



In the above figure, depending on the spec range, pass and fail regions are shaded with green and red patterns, respectively. The two white dotted bold lines that indicate the threshold value separate these regions. When you select a trace in the graph or in the trace legend, the thickness of the threshold line increases and the spec marker label background changes to its respective pass or fail color. Marker label displayed in each region shows the pass or fail status and the specification type and values. For example, the `pass:(range 95u 103u)` marker label displayed in the above graph shows that the specification type is `range` and the target range values are `95u` and `103u`. The results that are plotted in the region marked by this label meet the specification.

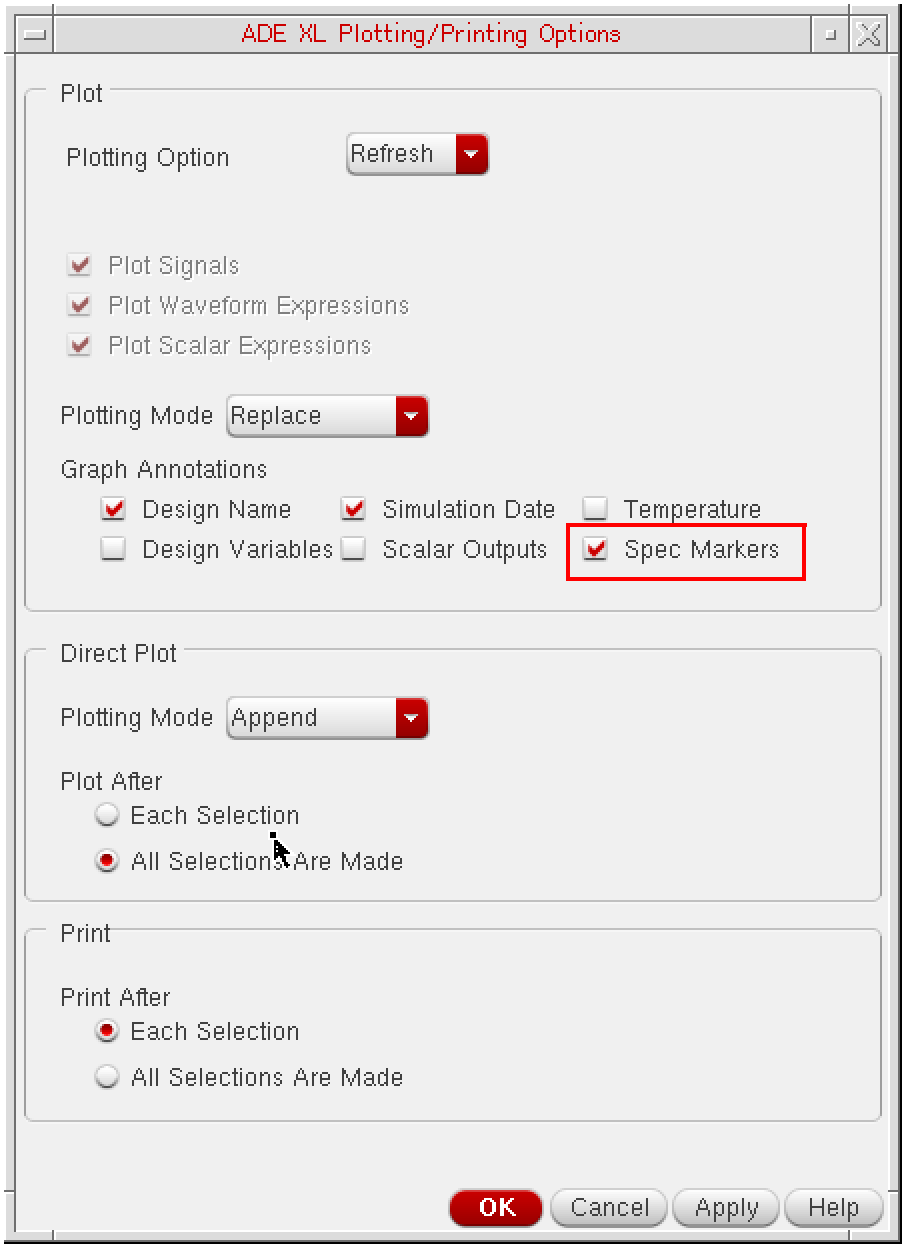
By default, the pass and fail labels appear near the threshold line. However, if the specification type is `range`, the pass label is displayed in the center as shown in Figure 14-15 on page 610.

To display the *Spec* column in trace legend, right-click the trace legend header and choose *Spec*.

Enabling Display of Spec Markers

To display spec markers, in the ADE XL Plotting/Printing Options form, select the *Spec Markers* graph annotation and click *Apply*.

Figure 14-16 ADE XL Plotting/Printing Options Form



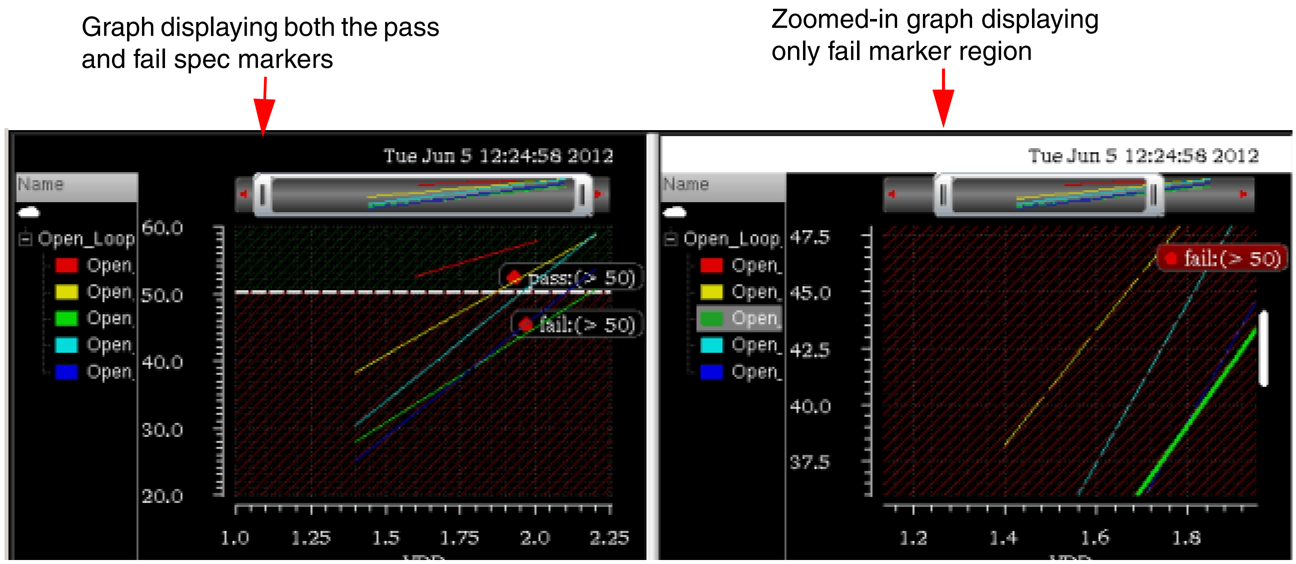
Note: By default, the *Spec Markers* option is deselected.

If you select the *Spec Markers* option and plot graphs, ADE XL checks for the presence of specifications for the result to be plotted. For the measurements to be plotted, if you have specified spec values in the ADE XL Outputs section, the tool shades the pass and fail regions in the graph.

Displaying Spec Markers on Zoomed-in Graph

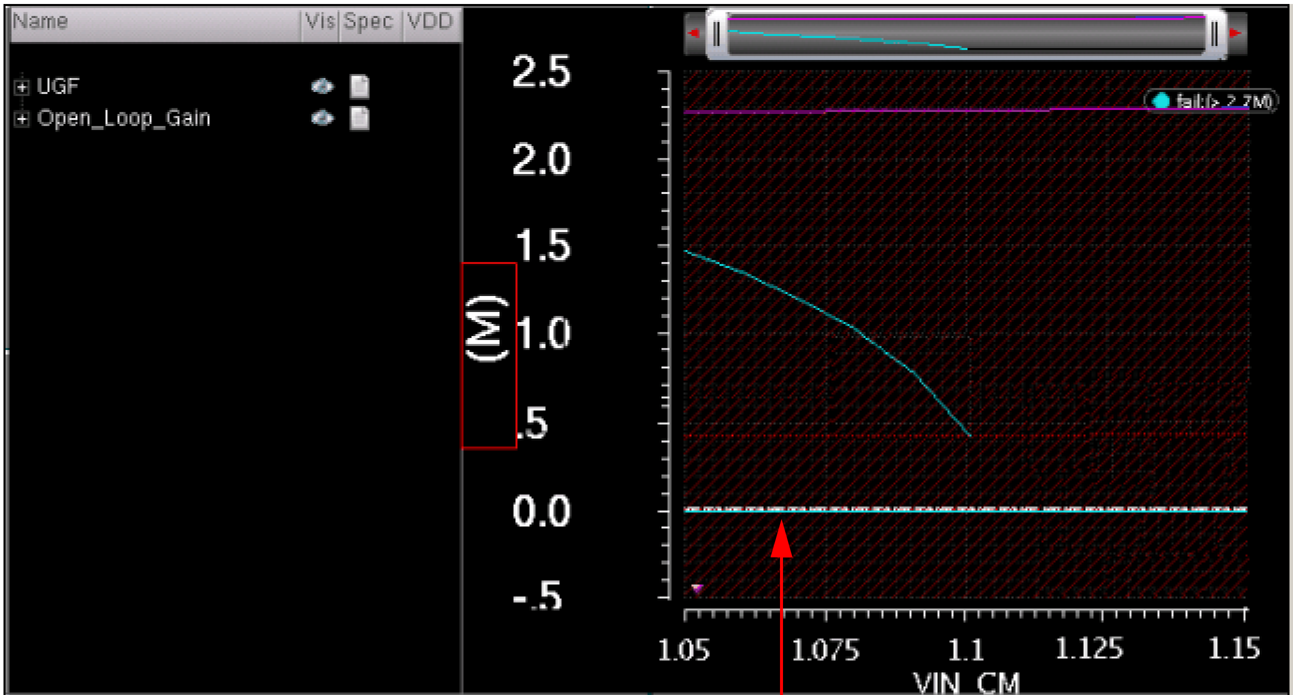
Spec markers are displayed on the graph only if the associated trace is visible. When you zoom in a graph, only the spec markers that fall in the zoomed-in graph portion are visible. This helps improve visibility when multiple spec markers are applied on multiple traces.

Figure 14-17 Spec Markers on Zoomed-in Graph



The figures below demonstrate an example when two traces are plotted in the same graph and share the same Y-axis. In this example, the trace `Open_Loop_Gain` has a huge Y-axis scale and the other overlaid trace, `UGF`, has a comparatively small Y-axis scale. Therefore, the spec markers for `Open_Loop_Gain` are not visible in the graph (as shown in Figure 14-17 on page 613). To display the hidden spec markers, you need to zoom in the graph until you get the small Y-axis scale.

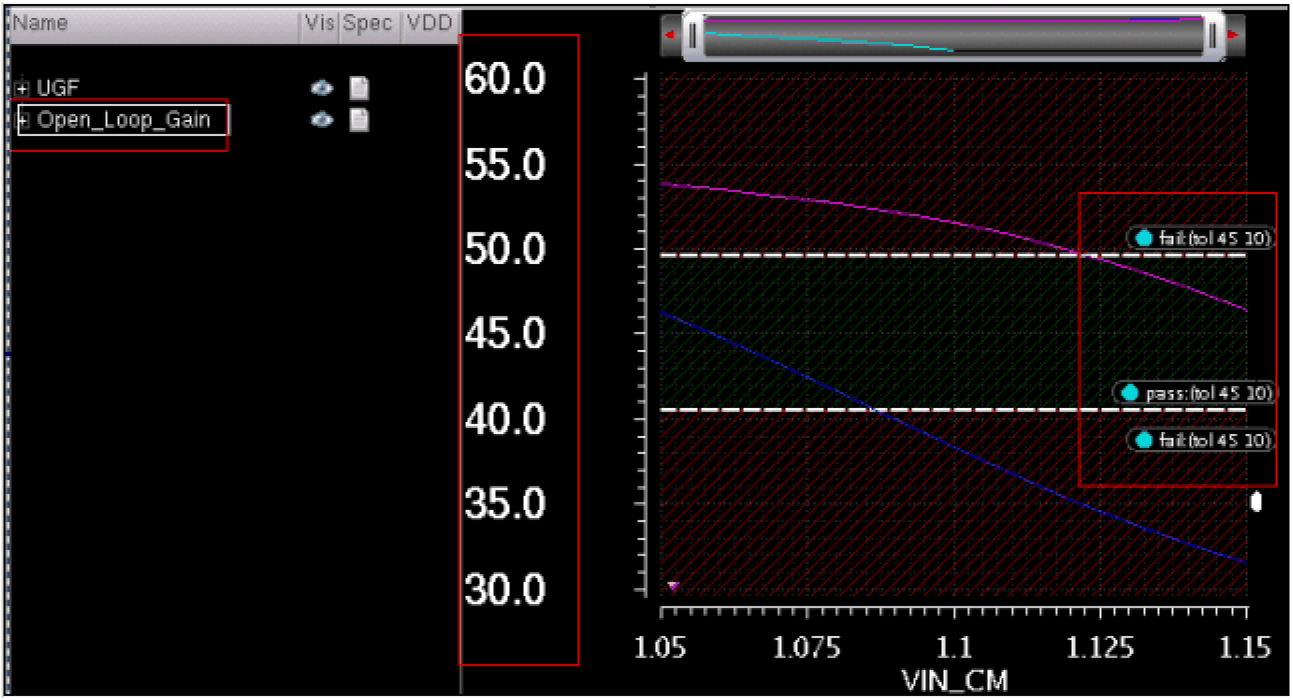
Figure 14-18 Example: Spec Markers on a Zoomed-in Graph



Spec marker pass/fail region is hidden for Open_Loop_Gain

When you zoom in the above displayed graph to `ymax=60` and `ymin=30`, the hidden spec markers become visible as shown in Figure 14-19 on page 615.

Figure 14-19 Example: Spec Markers on a Zoomed-in Graph



Spec Markers for Different Specification Types

Depending on the type of specification, ADE XL displays spec markers in different ways, as described in the following table.

Table 14-7 Spec Markers for Different Specification Types

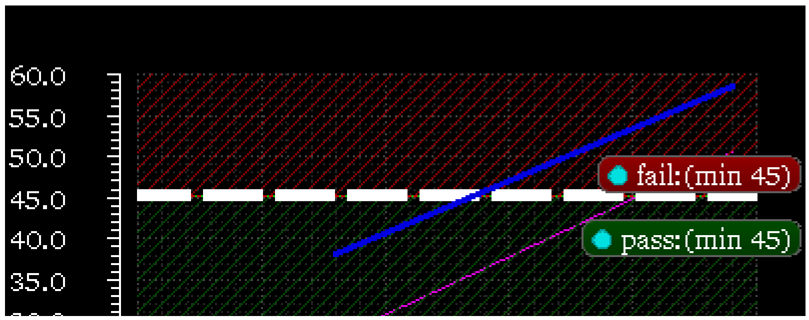
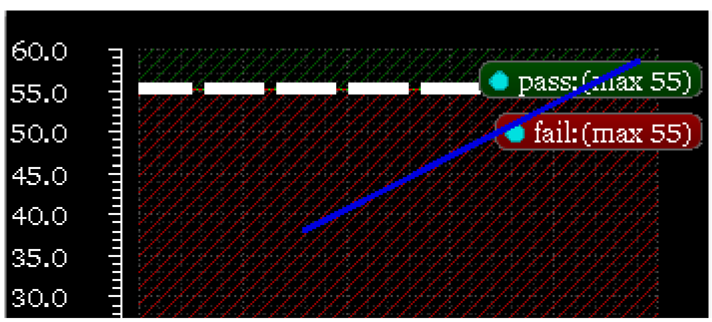
Specification Type	Spec Marker Display
minimum	Displays one horizontal marker line at the target minimum value. The region in which the results are below the minimum target is shaded with green and the region above it is shaded with red. 
maximum	Displays one horizontal marker line at the target maximum value. The region in which the results are greater than the maximum target is shaded with green and the region below it is shaded with red. 

Table 14-7 Spec Markers for Different Specification Types

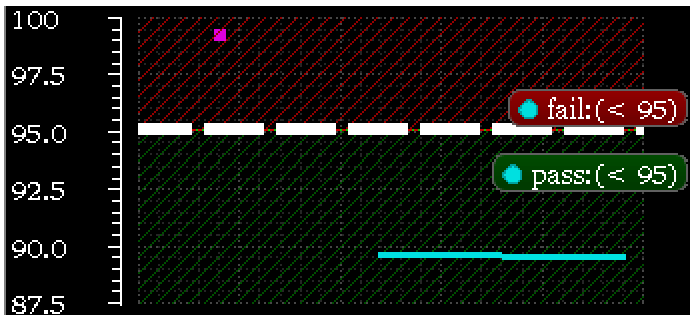
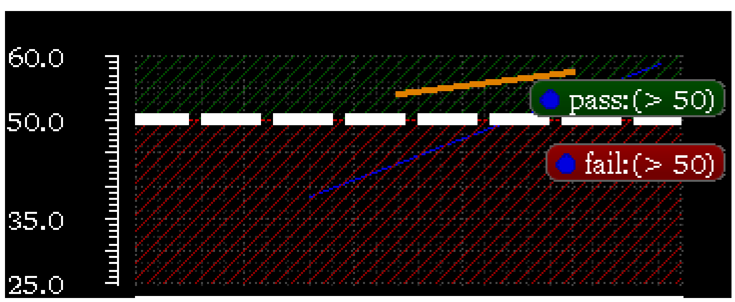
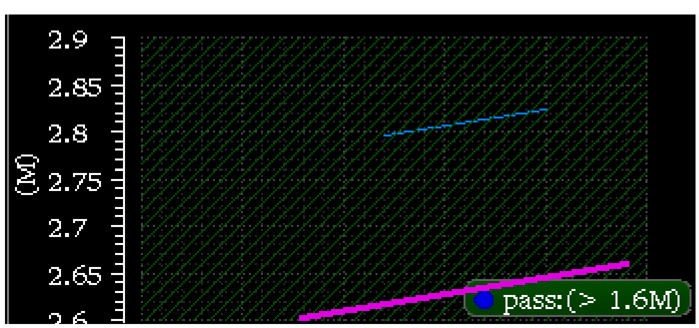
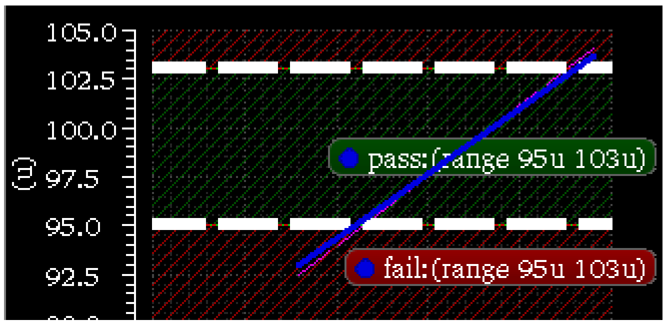
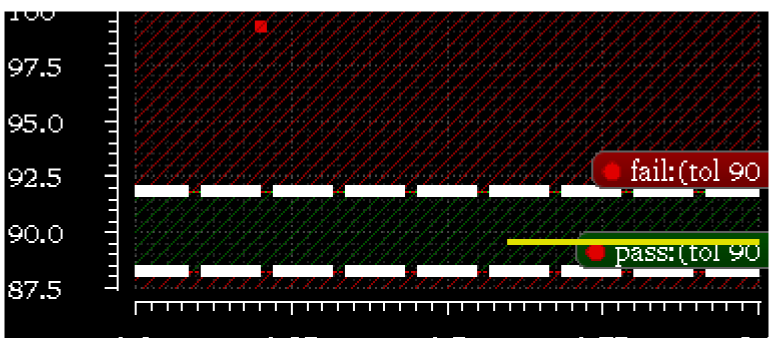
Specification Type	Spec Marker Display
<	Displays one horizontal marker line at the spec value. This specification requires that the results should be less than the spec value. Therefore, the region below the marker is shaded with green and the region above it is shaded with red. 
>	Displays one horizontal marker line at the spec value. This specification requires that the results should be greater than the spec value. Therefore, the region below the marker is shaded with red and the region above it is shaded with green.  If all the results appear in one region, the other region is not displayed, as shown in the figure below: 

Table 14-7 Spec Markers for Different Specification Types

Specification Type	Spec Marker Display
range	Displays two horizontal marker lines—one at the lower bound value and the other at the upper bound value. This specification type requires that results should be within the given range. Therefore, the region between the two marker lines is shaded with green and the region outside these lines is shaded with red. 
tol	Displays two horizontal marker lines—one at the lower end of the tolerance range and the other at the higher end. This specification type requires that results should be within the given tolerance range. Therefore, the region between the two marker lines is shaded with green and the region outside these lines is shaded with red. 
info	No marker is displayed for this specification type.

Important

While displaying spec markers, ADE XL does not consider overridden specifications for corners. If any overridden specification exists, spec markers are displayed by considering only the global specifications.

Changing Properties of Spec Markers

You can change the display of spec markers by setting their properties in the Virtuoso Visualization and Analysis XL graph. For example, you can change the shading color or you can choose to show or hide the markers in the pass or fail region.

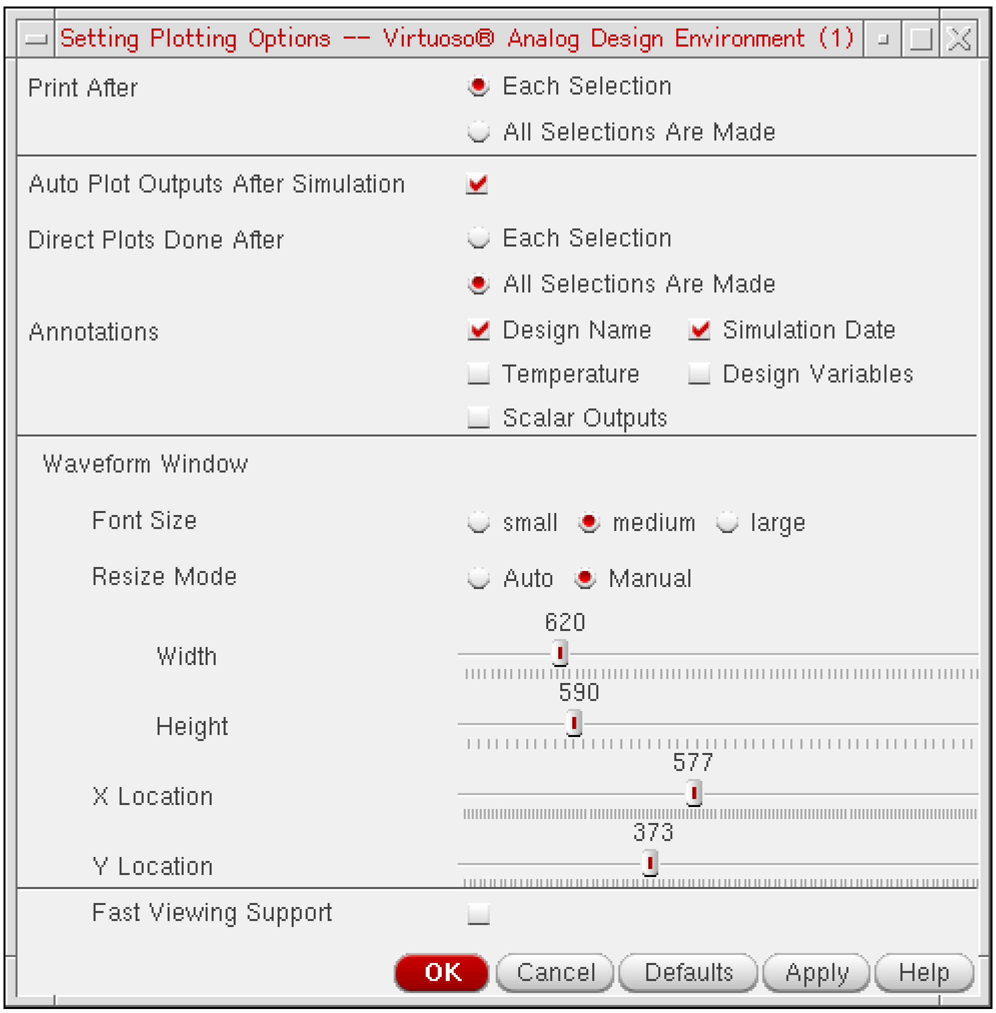
For more details, refer to the Changing Spec Marker Properties section in the *Virtuoso Visualization and Analysis XL User Guide*.

Setting Plotting Options for Specific Tests

You can use the Setting Plotting Options form to set various options related to printing and plotting for specific tests. The options you specify in this form override the options specified in the ADE XL Setting Plotting Options form. For more information about the ADE XL Setting Plotting Options form, see Setting Default Plotting Options for All Tests on page 587.

1. On the Outputs Setup tab of the Outputs pane, right-click a test name and choose *Printing/Plotting Options*.

   The Setting Plotting Options form appears.



2. Specify plotting options.

   See the following topics for assistance:

   ❑ Specifying When to Print Results on page 621

- ❑ Specifying Whether to Plot Automatically After Simulation on page 621
- ❑ Specifying When to Plot Direct Plot Results on page 622
- ❑ Specifying Annotations for the Graph Window on page 592
- ❑ Specifying Waveform Window Configuration Information on page 622
- ❑ Enabling Fast Waveform Viewing Format for PSF Output on page 623

3. When you are finished specifying options, click *OK*.

Specifying When to Print Results

To specify when to print results to the Results Display Window, do the following:

1. Select one of the following *Print After* options:
   - ❑ *Each Selection* – Print results after each item you select on the schematic.
   - ❑ *All Selections Are Made* – Print results after you select all items and press *Esc* to end selection mode.
2. Click *Apply*.

Return to main procedure.

Specifying Whether to Plot Automatically After Simulation

To specify that you want to plot outputs automatically after the simulation finishes, do the following:

1. Select the *Auto Plot Outputs After Simulation* check box.
2. Click *Apply*.

To specify that you do not want to plot outputs automatically after the simulation finishes, do the following:

1. Deselect the *Auto Plot Outputs After Simulation* check box.
2. Click *Apply*.

Return to main procedure.

Specifying When to Plot Direct Plot Results

To specify when to plot results you select from your schematic when you use the *Results – Direct Plot* submenu, do the following:

1. Select one of the following *Direct Plots Done After* options:
   - ❑ *Each Selection* – Plot results after each item you select on the schematic.
   - ❑ *All Selections Are Made* – Plot results after you select all items and press *Esc* to end selection mode.
2. Click *Apply*.

Return to main procedure.

Specifying Waveform Window Configuration Information

You can specify waveform window configuration information such as font size, window height and width, X and Y location.

To specify waveform window configuration information, do the following:

1. In the *Waveform Window* group box, use the sliders to change one or more of the following settings:
   - ❑ *Font Size* – Sets the font size for the text on the waveform window.
   - ❑ *Width* – Sets the initial width of the waveform window.
   - ❑ *Height* – Sets the initial height of the waveform window.
   - ❑ *X Location* – Sets the initial X location of the upper left corner of the waveform window: a higher number places the window farther to the right on your display.
   - ❑ *Y Location* – Sets the initial Y location of the upper left corner of the waveform window: a higher number places the window lower on your display.
2. Click *Apply*.

Tip

To preserve these settings for future design sessions, do the following:

a. In the Command Interpreter Window (CIW), choose *Options – Save Defaults*.

   The Save Defaults form appears.

b. Click *OK*.

Return to main procedure.

Enabling Fast Waveform Viewing Format for PSF Output

Using the PSF output in the fast waveform viewing format, Virtuoso Visualization and Analysis can render extremely large datasets (where signals have a large number of data points, for example 10 million) within seconds.

To enable the fast waveform viewing format for PSF output, do the following:

- Select the *Fast Viewing Support* check box.

Return to main procedure.

Plotting Results

You can plot the results that appear on the Results tab of the Outputs pane. The plots are displayed in a waveform window.

Important

Starting with IC6.1.5 ISR6 release, the structure of the psf directory where simulation results are saved has been enhanced for single point runs. As a result, you may observe the following changes:

i) Similar to multiple point runs, the simulation results for a single point run are now also saved under the top level psf directory. That is, the results are saved in:
- Interactive.<n>/psf/<test_name>/psf, which is the top level psf directory
- Interactive.<n>/1/<test_name>/psf, which is the psf directory for the single point

ii) When you set context from Result Browser to the top level psf directory and evaluate a scalar expression in Calculator, the result is returned as a waveform object. However, when you set context to the results directory of the single point run, the result of a scalar expression is returned as a scalar value.

iii) When you plot results of a single point run, if no global variables or parameters were used, the trace legend of the graph shows the Design Point column. If global variables or parameters were used to run the simulation for that point, name of last variable or parameter in the global list is shown in the trace legend.

For more information, see the following topics:

- Plotting Across All Points on page 625
- Plotting Across All Corners and Sweeps on page 625
- Plotting Across Corners on page 626
- Plotting Selected Points on page 627
- Plotting Outputs from the Specified Plot Set on page 628
- Other Plotting Methods on page 628

Tip

You need not wait for the simulation run to complete before plotting results. You can plot the results for completed data points and corners even when the simulation run is in progress.

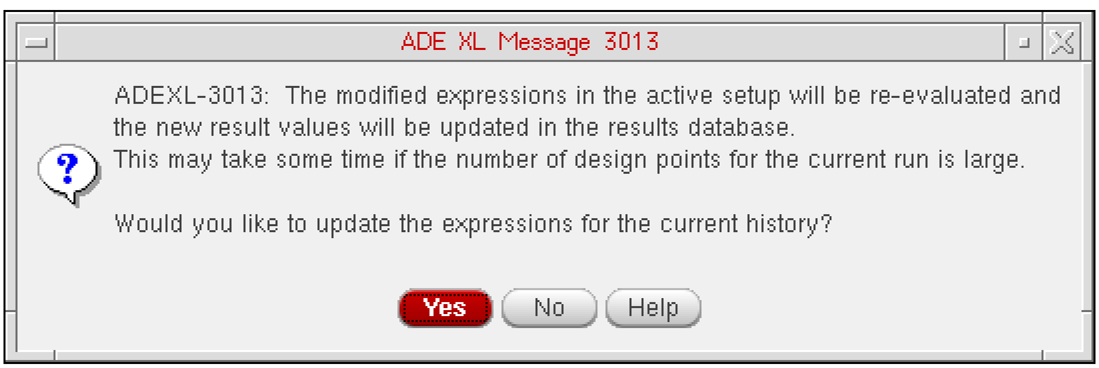
Plotting Across All Points

To plot across all points, do the following:

1. Select the plot mode. For more information, see Selecting the Plot Mode on page 585.
2. On the Results tab of the Outputs pane, click the button.

   The waveforms appear in a waveform window.

   Note: If you click the button after modifying measurement expressions or specifications in the Outputs Setup tab, the following message box appears. Click *Yes* to update the results displayed in the Results tab based on the new or modified expressions and specifications and then plot the results. Click *No* to ignore the changes in expressions and specifications and plot the results.



Plotting Across All Corners and Sweeps

To plot waveforms across all corners and sweeps for a signal or waveform expression, do the following:

1. Select the plot mode. For more information, see “Selecting the Plot Mode” on page 585.
2. Do one of the following:

- ❑ Double-click on a test name in the *Test* column or a signal or waveform expression name in the *Output* column of the Results tab.
- ❑ Right-click on a test name in the *Test* column or a signal or waveform expression name in the *Output* column of the Results tab and choose *Plot All*.

The waveforms are displayed in a waveform window.

Plotting Across Corners

To plot waveforms across all corners for a signal or waveform expression, do the following:

1. Select the plot mode. For more information, see "Selecting the Plot Mode" on page 585.
2. Right-click on any of the following on the Results tab and choose *Plot Across Corners*:
    - ❑ A test name in the *Test* column
    - ❑ A signal or waveform expression name in the *Output* column
    - ❑ In the *Nominal* column or in the column for a corner

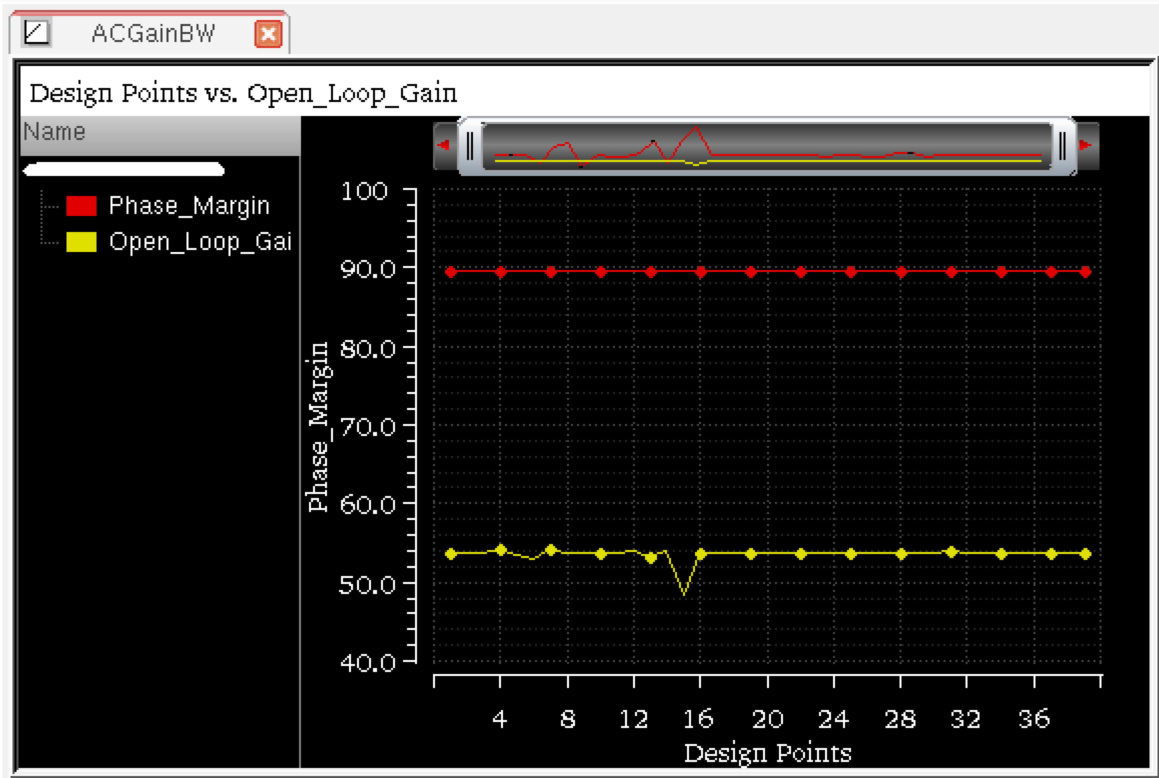
    The waveforms are displayed in a waveform window.

Plotting Across Design Points

To plot waveforms across all design points for a signal or waveform expression, do the following:

1. Select the plot mode. For more information, see "Selecting the Plot Mode" on page 585.
2. On the Results tab, right-click on the *Nominal* column or in the column for a corner and choose *Plot Across Design Points*.

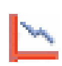
    The waveforms displayed in a waveform window show results plotted across all design points, as shown below.



Note that in the above figure, graphs have been plotted after setting the `Append` plotting mode.

Plotting Selected Points

To plot waveforms selectively, select the plot mode (see "Selecting the Plot Mode" on page 585) and then do one of the following:

- Right-click a signal (indicated by the  icon) or a waveform expression in the *Nominal* column or in the column for a corner and choose *Plot*.

  To plot multiple signals or waveform expressions, hold down the *Shift* key (for contiguous selection) or the *Ctrl* key (for noncontiguous selection) and click the next signal or waveform expression to add it to the selection set. Right-click and choose *Plot*.

  Note: You cannot plot multiple signals or waveform expressions for Monte Carlo simulation results.

- Double-click on a signal or waveform expression in the *Nominal* column or in the column for a corner.

The waveforms for the selected signals and waveform expressions appear in a waveform window.

Plotting Outputs from the Specified Plot Set

To plot outputs from the specified plot set, do the following:

1. Select the plot mode. For more information, see Selecting the Plot Mode on page 585.
2. Do one of the following on the Results tab of the Outputs pane:
   - ❑ To plot outputs across all corners and sweeps for a signal or waveform expression, right-click on a test name in the *Test* column or a signal or waveform expression name in the *Output* column and choose *Plot Outputs*.
   - ❑ To plot outputs for a specific point, right-click on a signal or waveform expression in in the *Nominal* column or in the column for a corner and choose *Plot Outputs*.
3. Select one of the following from the submenu that appears:

Transient	Plots the transient response for each node
AC	Plots the AC response for each node
Noise	Plots the squared noise voltage for each node
DC	Plots the DC sweep response for each node
Expressions	Plots the waveforms for expressions you define on the Outputs Setup tab of the Outputs pane.

The outputs that you specified for plotting appears in a plot window according to the plotting mode you selected.

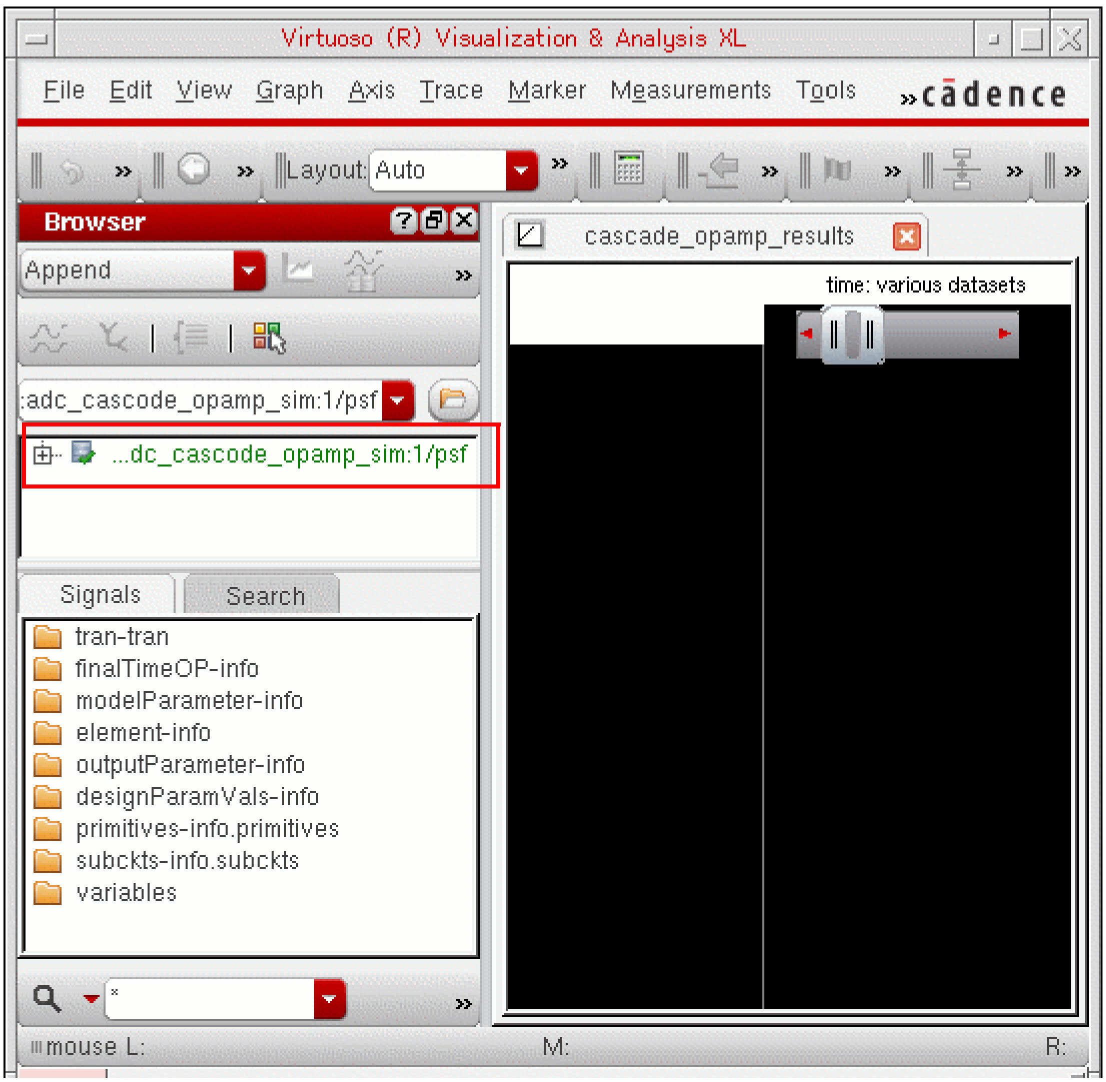
Other Plotting Methods

You can also use any of the following methods to plot selected simulation results in a waveform window:

- On the Results tab of the Outputs pane, right-click any of the following and choose *Open Results Browser* to load the selected results in the Results Browser:
  - ❑ Right-click a test name in the *Test* column or a signal or waveform expression name in the *Output* column to plot across all the corners for the signal or waveform expression.

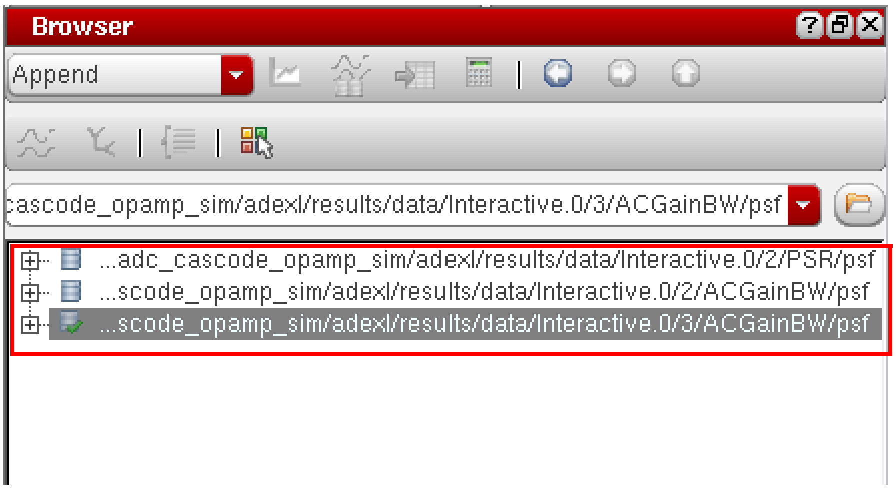
- ❑ Right-click one or more signal or waveform expressions in the *Nominal* column to plot across the nominal corner.
- ❑ Right-click one or more signal or waveform expressions in a corner column to plot across the corner.

When you load a single result in the Results Browser, the Virtuoso Visualization and Analysis XL window is opened. The selected results are opened in the Results Browser and context is set to the psf directory of these results. This is indicated by a green check mark on the name of results, as shown in the following figure.



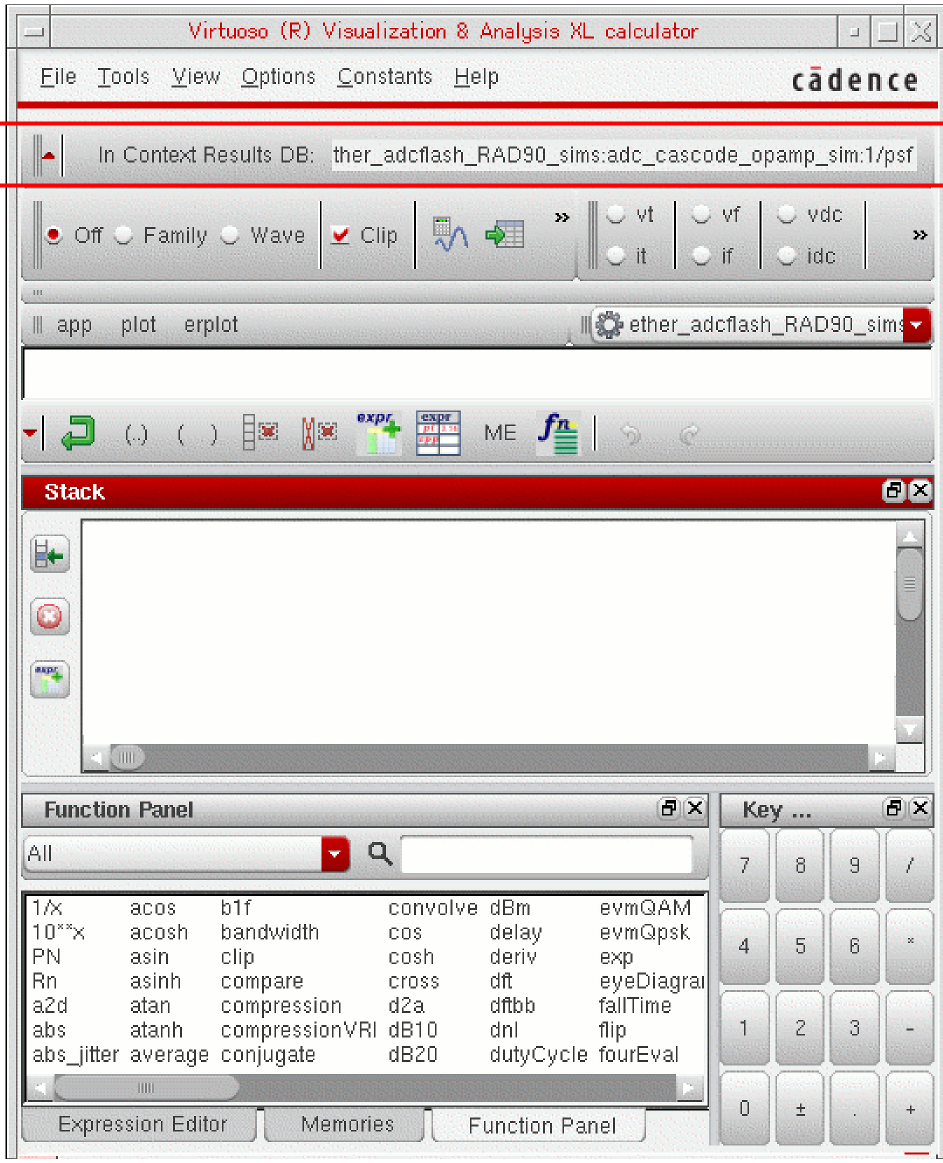
Navigate to a node that contains waveform data and right-click to select a plot mode (such as *Append*, *Replace*, *New SubWin*, *New Win*). For more details, refer to *Virtuoso Visualization and Analysis User Guide*.

When you load multiple results in the Results Browser, the psf directories of all the points are opened in the sequence of selection. The context is set to the psf that was selected last, as shown in the following figure.



- On the Results tab of the Outputs pane, right-click any of the following and choose *Direct Plot* for a submenu of functions for plotting selected nets and terminals from your schematic.
    - Right-click a test name in the *Test* column or a signal or waveform expression name in the *Output* column to direct plot across all the corners for a signal or waveform expression.
    - Right-click a signal or waveform expression in the *Nominal* column to direct plot across the nominal corner.
    - Right-click a signal or waveform expression in a corner column to direct plot across the corner.
- On the Results tab of the Outputs pane, right-click a test name in the *Test* column or a signal or waveform expression name in the *Output* column or a single data point and choose *Open Calculator*. The Virtuoso Visualization and Analysis XL calculator window

is opened. The current context in this window is set to the psf directory of selected results data.



Now, you can create expressions using the data point and plot outputs from calculator.

See also

- Chapter 11, "Selecting Data to Save and Plot"
- "Selecting the Plot Mode" on page 585

Using Direct Plot

You can use the direct plotting feature to plot a function from a set of one or more nets or terminals that you select on your schematic.

Tip

You need not wait for the simulation run to complete before using direct plotting. You can use direct plot for completed data points and corners even when the simulation run is in progress.

To use direct plot, do the following:

1. Select the plot mode. For more information, see "Selecting the Plot Mode" on page 585.

2. On the Results tab of the Outputs pane, right-click on any of the following and choose *Direct Plot*.

   ❑ Right-click on a test name in the *Test* column or a signal or waveform expression name in the *Output* column to direct plot across all the corners for a signal or waveform expression.

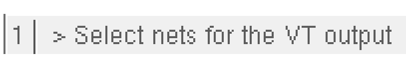
   ❑ Right-click on a signal or waveform expression in the *Nominal* column to direct plot across the nominal corner.

   ❑ Right-click on a signal or waveform expression in a corner column to direct plot across the corner.

   A submenu of functions appears. Only those functions that apply to the current results are available for selection.

3. Select a function.

   If you have the prompt line turned on, you can read the prompt text for a hint about what you need to select.



4. In the schematic window, select one or more nets or terminals to plot using the function you selected.

5. Press *Esc* when you are done selecting.

   The selected nets or terminals are plotted in a waveform window according to the function you selected. If a waveform window was not already open, then one appears. If a waveform window was already open, then it appears in the foreground.

Note: If the *Add To Outputs* check box is selected in the Direct Plot Form, expressions are added for the plotted results in the Outputs Setup tab of the Outputs pane. Re-evaluate the expressions to display the updated results in the Results tab of the Outputs pane. For more information about re-evaluating expressions, see Re-evaluating Expressions and Specifications on page 650.

Here is a table of direct plot functions.

Direct Plot Functions

Function	Description
Main Form	Opens the Direct Plot Form for specifying plotting mode, analysis, function, and modifier (see "Using the Direct Plot Main Form" in the *Virtuoso Analog Design Environment L User Guide*).
Transient Signal	Transient voltage or current waveforms
Transient Minus DC	Transient voltage or current waveforms without the DC offset
Transient Sum	Multiple signals added together and plotted; you are prompted for the signals
Transient Difference	Two signals subtracted (sig1- sig2) and plotted; you are prompted for two signals
AC Magnitude	AC voltage or current gain waveform
AC dB10	The magnitude on a decibel scale 10log(V1)
AC dB20	The magnitude of selected signals on a decibel scale 20log(V1)
AC Phase	AC voltage or current phase waveform
AC Magnitude & Phase	The db20 gain and phase of selected signals simultaneously
AC Gain & Phase	The differences between two magnitudes and two phases; you are prompted for two signals 20log(V2)-20log(V1) which is equivalent to 20log(V2/V1)
Equivalent Output Noise	Output noise voltage or current signals selected in the analysis form; the curve plots automatically and does not require selection
Equivalent Input Noise	Input noise waveform, which is the equivalent output noise divided by the gain of the circuit

Direct Plot Functions, *continued*

Function	Description
Squared Output Noise	Squared output noise voltage or current signals selected in the analysis form; the curve plots automatically and does not require selection
Squared Input Noise	Input noise waveform, which is the equivalent output noise divided by the gain of the circuit squared
Noise Figure	Noise figure of selected signals according to the input, output, and source resistance
DC	DC sweep voltage or current waveform

Printing Results

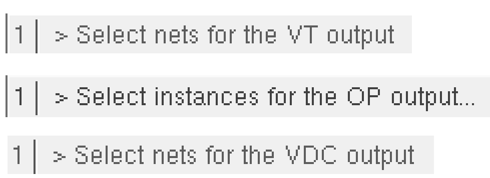
To print text results and reports to the Results Display Window, do the following:

1. On the Results tab of the Outputs pane, right-click a test name and choose *Print*.

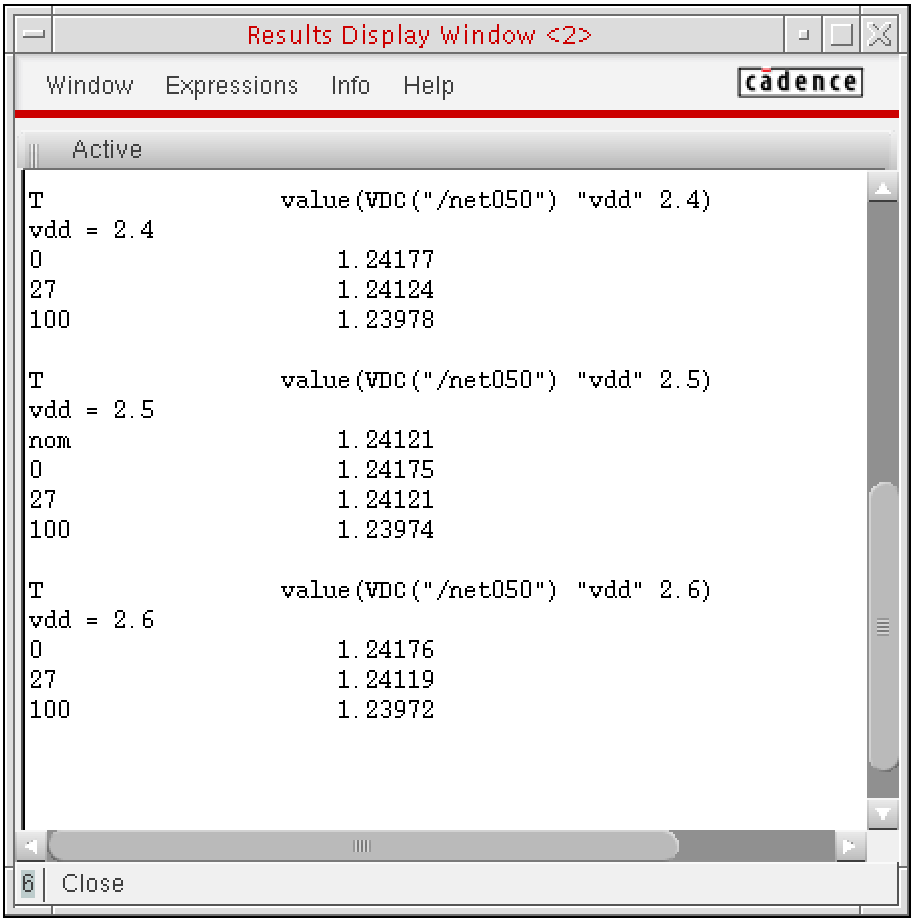
   A submenu of functions appears.
   Only those functions that apply to the current results are available for selection.

2. Select a function.

   If you have the prompt line turned on, you can read the prompt text for a hint about what you need to select.



The Results Display Window appears.



3. In the schematic window, select one or more nets or terminals to plot using the function you selected.

The values for the selected nets or terminals appear in the Results Display Window according to the function you selected. If a Results Display Window was not already open, then one appears. If a Results Display Window was already open, then it appears in the foreground.

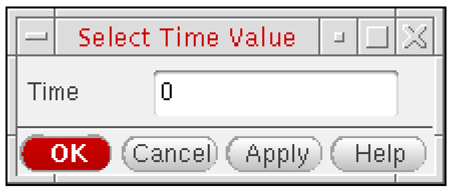
Note: See also "Specifying When to Print Results" on page 621.

Here is a table of print functions.

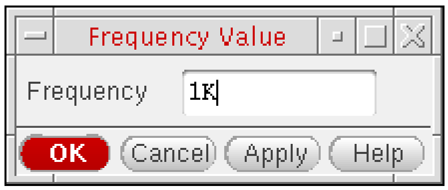
Print Functions

Function	Description
DC Node Voltages **Note:** This function is available following a DC analysis.	Print DC node voltages of selected nodes **1.** In the schematic window, select one or more nodes. **2.** Press *Esc* when you are finished selecting.
DC Operating Points **Note:** This function is available following a DC analysis.	Print DC operating points of selected instances **1.** In the schematic window, select one or more instances. If you select an instance that is a subcircuit definition, the program prints operating points for all devices in the subcircuit. **2.** Press *Esc* when you are finished selecting. **Note:** It may take some time to search for all instances in a subcircuit definition. To disable this feature, you can set the following environment variable in your `.cdsenv` file: `asimenv.printing  printInlines  boolean  nil`
Model Parameters	Print model parameters of selected instances **1.** In the schematic window, select one or more instances. If you select an instance that is a subcircuit definition, the program prints model parameters for all devices in the subcircuit. **2.** Press *Esc* when you are finished selecting. **Note:** It may take some time to search for all instances in a subcircuit definition. To disable this feature, you can set the following environment variable in your `.cdsenv` file: `asimenv.printing  printInlines  boolean  nil`

Print Functions, *continued*

Function	Description
Transient Node Voltages **Note:** This function is available following a transient analysis.	Print transient node voltages of selected nodes The Select Time Value form appears when you select this function.  **1.** In the *Time* field, type the time value for which you want to print transient node voltages. **2.** Click *OK*. **3.** In the schematic window, select one or more nodes. **4.** Press *Esc* when you are finished selecting.
Transient Operating Points **Note:** This function is available following a transient analysis.	Print final transient operating points of selected items **1.** In the schematic window, select one or more instances or nodes. If you select an instance that is a subcircuit definition, operating point values for all devices in the subcircuit appear in the Results Display Window. **2.** Press *Esc* when you are finished selecting. **Note:** It may take some time to search for all instances in a subcircuit definition. To disable this feature, you can set the following environment variable in your `.cdsenv` file: `asimenv.printing  printInlines  boolean  nil`
S-Parameter	Print S-parameter data

Print Functions, *continued*

Function	Description
Noise Parameters **Note:** This function is available following a noise analysis.	Print noise parameters of selected instances The Select Frequency Value form appears when you select this function. The default frequency is 1K.  **1.** In the *Frequency* field, type the frequency value for which you want to print noise parameters. **2.** Click *OK*. **3.** In the schematic window, select one or more instances. **4.** Press *Esc* when you are finished selecting.
Noise Summary **Note:** This function is available following a noise analysis.	Opens the Noise Summary form so you can print noise contributions of selected instances See "Printing Noise Summary Information" on page 639 for more information.
DC Mismatch Summary **Note:** This function is available following a `dcmatch` analysis.	Opens the Dcmatch Summary form so you can print DC mismatch information for your design See "Printing DC Mismatch Summary" on page 646 for more information.
Stability Summary **Note:** This function is available following a stability analysis.	Print phase margin and gain margin results for every combination of sweep variable values
Pole Zero Summary **Note:** This function is available following a pole-zero analysis.	Opens the Pole-Zero Summary form so you can print pole-zero information for your design See "Printing Pole Zero Summary" on page 648 for more information.

Print Functions, *continued*

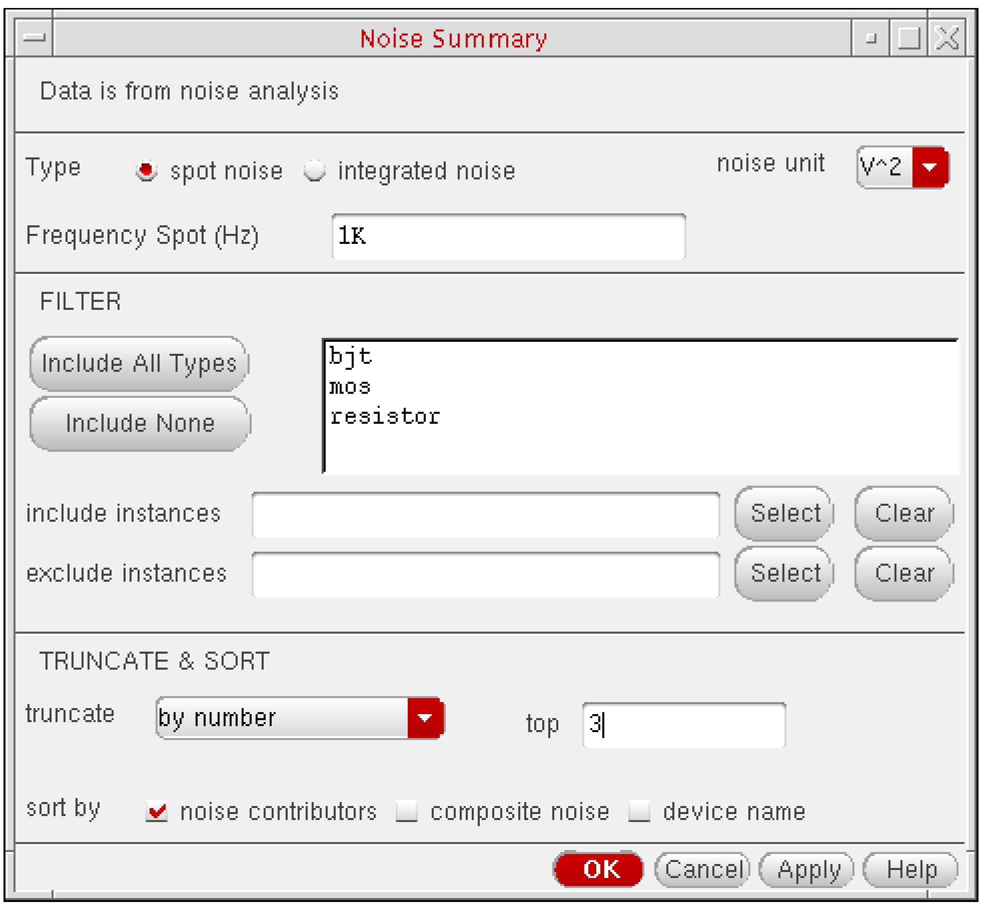
Function	Description
Sensitivities **Note:** This function is available following a sensitivity analysis.	Print sensitivities **1.** In the schematic window, select one or more nets or ports. **2.** Press *Esc* when you are finished selecting.

Printing Noise Summary Information

To print noise contributions of selected instances, do the following:

1. On the Results tab of the Outputs pane, right-click a test name and choose *Print – Noise Summary*.

   The Noise Summary form appears.



2. Choose one of the following noise summary types:
   - *spot noise* – Type a frequency in the *Frequency Spot (Hz)* field.
   - *integrated noise* – Specify a range and weighting option.
3. In the *FILTER* group box, specify options to include or exclude devices types or instances.
4. (Optional) In the *TRUNCATE & SORT* group box, specify options to truncate and sort your noise summary data.
5. Click *OK*.

   The noise summary information you specified appears in the Results Display Window.

See also "Controlling Precision of Printed Noise Data" on page 645.

Specifying Options for Integrated Noise

When you select *integrated noise*, the *From (Hz)* and *To (Hz)* fields become active and you can specify a *weighting* option.

To specify options after selecting *integrated noise*, do the following:

1. In the *From (Hz)* field, type a starting value for the frequency range for the integration.
2. In the *To (Hz)* field, type an ending value for the frequency range for the integration.
3. Select one of the following *weighting* options:
   - *flat* – The program integrates over the original unweighted waveform.
   - *from weight file* – You can specify a file containing weight factors to apply to the noise contributions of particular frequencies prior to integrating. The file must contain one of the following entries as the first line: `db`, `mag`, `dbl`, `DB`, `MAG`, `DBL`. Each additional line must contain a pair of X and Y values. All the pairs together must define a function. For example:

     ```
     mag
     1          .001641
     60         .001641
     100        .007499
     200        .05559
     ```
4. Return to "Printing Noise Summary Information" on page 639.

Specifying Device Types and Instances to Include or Exclude

Device types in your design appear in the list box in the *FILTER* group box.

To specify which device types to include and which to exclude from the noise summary, do the following:

1. To include all device types in the summary, click *All Types*.
2. (Optional) To exclude individual device types, hold down the *Ctrl* key and click each one.

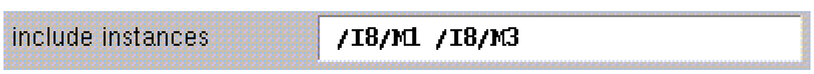
Alternatively, you can do the following:

1. To exclude all device types from the summary, click *None*.
2. Select each device type you want to include in the summary:
   - ❑ You can hold down the *Shift* key to select more than one contiguous device type.
   - ❑ You can hold down the *Ctrl* key to select more than one noncontiguous device type.

To specify instances to include in the noise summary, do the following:

1. To the right of the *include instances* field, click *Select*.
2. In the schematic window, select one or more instances.

   Each instance path appears in the *include instances* field.

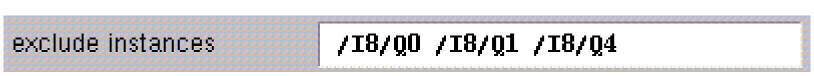


3. Press *Esc* when you are finished selecting.

To specify instances to exclude from the noise summary, do the following:

1. To the right of the *exclude instances* field, click *Select*.
2. In the schematic window, select one or more instances.

   Each instance path appears in the *exclude instances* field.



3. Press *Esc* when you are finished selecting.

Truncating Noise Summary Data

From the *TRUNCATE & SORT* group box on the Noise Summary form, you can select a truncation option to limit the number of noise contributors that appear in the Results Display Window when you click *Apply* or *OK*.

To specify no truncation of noise data, do the following:

➤ Select *none*.

All noise contributors appear in the noise summary report.

To limit the number of noise contributors that appear in the summary, do the following:

1. Select *by number*.
2. In the *top* field, type the number of highest noise contributors you want to see.

   The program reports only that number of noise contributors.

To limit the report to only those devices that contribute a certain percentage of the total noise, do the following:

1. Select *by rel. threshold*.
2. In the *noise %* field, type the minimum percentage noise contribution threshold.

   The program reports only those devices that contribute at least the minimum percentage of the total noise.

To limit the report to only those devices that contribute a minimum level of noise, do the following:

1. Select *by abs. threshold*.
2. In the *noise value* field, type the minimum noise contribution threshold.

   The program reports only those devices that contribute at least the minimum noise value.

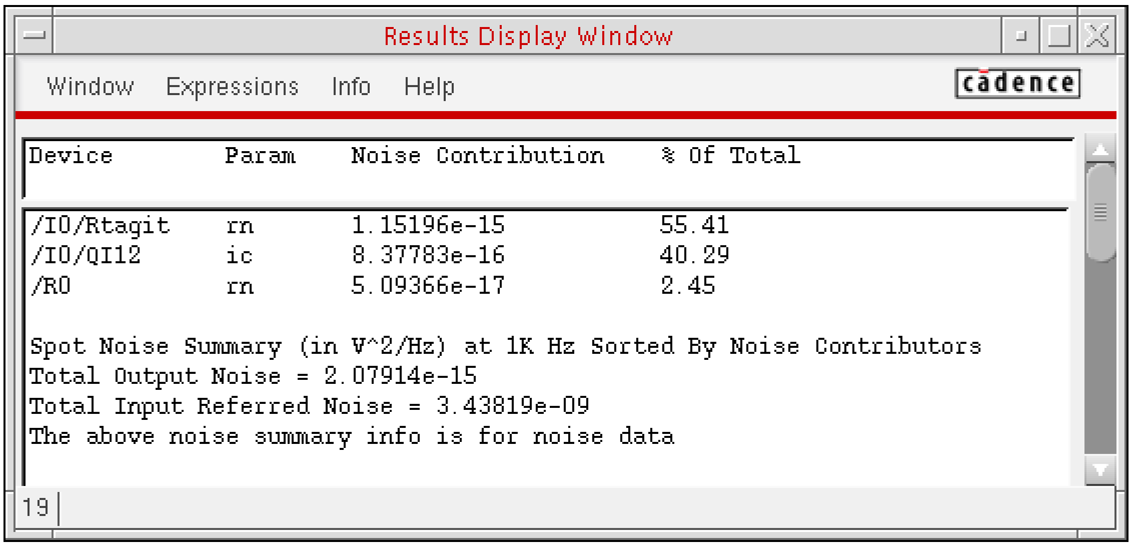
Sorting Noise Summary Data

From the *TRUNCATE & SORT* group box on the Noise Summary form, you can sort the list of devices that the program reports by noise contributors (highest to lowest), composite noise (highest to lowest), or device name (alphabetical from A to Z). The report appears in the Results Display Window when you click *Apply* or *OK*.

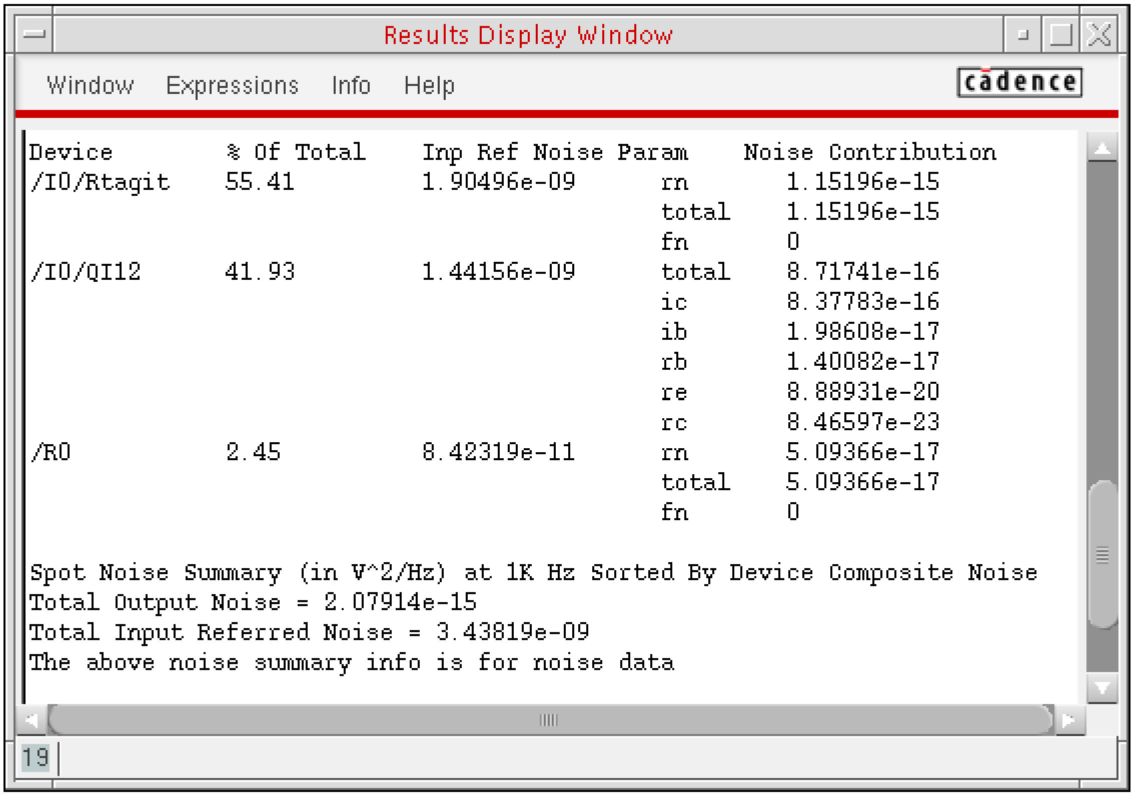
To request one or more sorted lists of noise contributors in your noise summary report, do the following:

➤ Select one or more of the following *sort by* options:

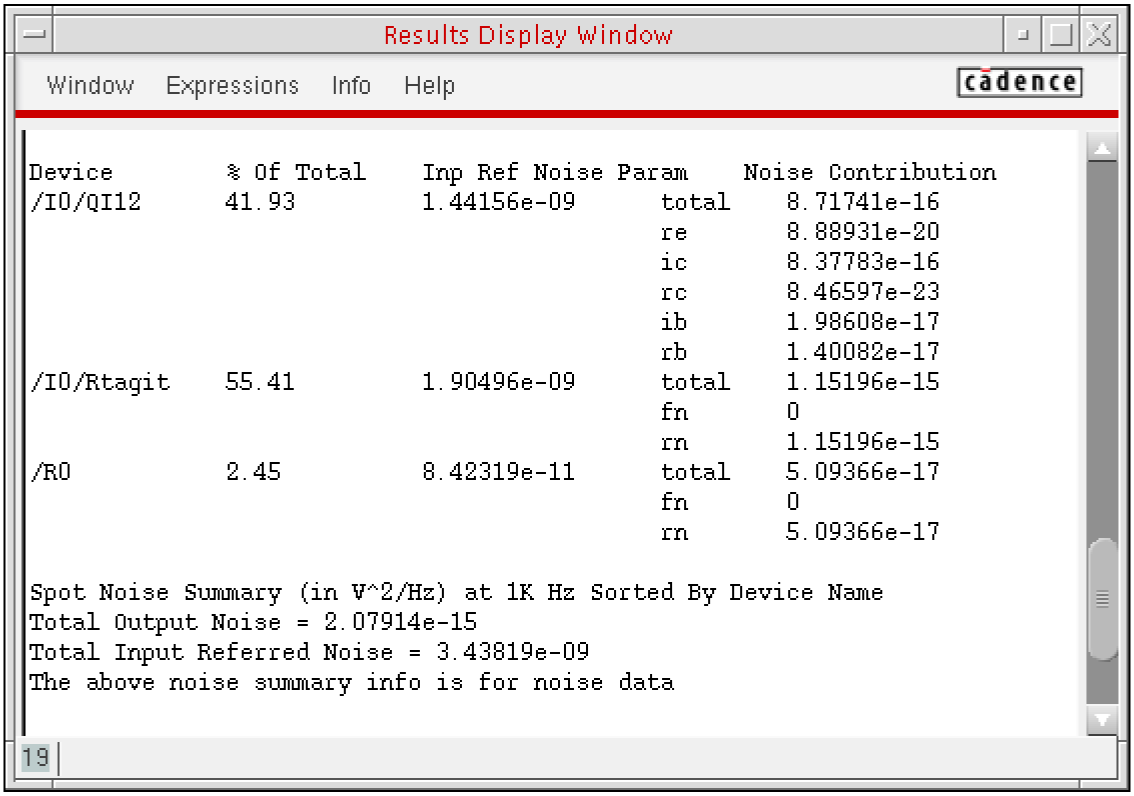
- ❑ *noise contributors* – Sorts devices by noise contribution, highest to lowest.



- ❑ *composite noise* – Sorts devices by composite noise contribution, highest to lowest.



- ❑ *device name* – Sorts devices by device name in alphabetical order from A to Z.



You can select one, two, or three check boxes. The number of boxes you select determines the number of lists that appear in your Results Display Window when you click *Apply* or *OK*.

Controlling Precision of Printed Noise Data

You can control the precision of them noise data that you print by setting the following variable in your `.cdsenv` file:

```
asimenv.noiseSummary digits int numberOfDigits
```

where *`numberOfDigits`* is the number of digits to print.

For example, to specify ten digits of precision, use the following setting:

```
asimenv.noiseSummary digits int 10
```

The default value for this variable is `6`.

Alternatively, you can set this value for the current session using the following command in your CIW:

```
envSetVal("asimenv.noiseSummary" "digits" 'int 10)
```

You can specify the number of digits to use when printing relative noise contributions by setting the following variable in your `.cdsenv` file:

```
asimenv.noiseSummary percentDecimals int numberOfDigits
```

where *`numberOfDigits`* is the number of digits to print.

For example, to specify four digits for relative contributions, use the following setting:

```
asimenv.noiseSummary percentDecimals int 4
```

The default value for this variable is `2`.

Alternatively, you can set this value for the current session using the following command in your CIW:

```
envSetVal("asimenv.noiseSummary" "percentDecimals" 'int 4)
```

You can control the precision of your printed results using the `aelPushSignifDigits` SKILL function as follows:

```
aelPushSignifDigits(numDigits)
```

where *`numDigits`* is the number of digits of precision you want.

Example

```
aelPushSignifDigits(4)
rn              37.9322e-18    fn             0              total
37.9322e-18
aelPushSignifDigits(8)
rn              37.932238e-18  fn             0              total
37.932238e-18
```

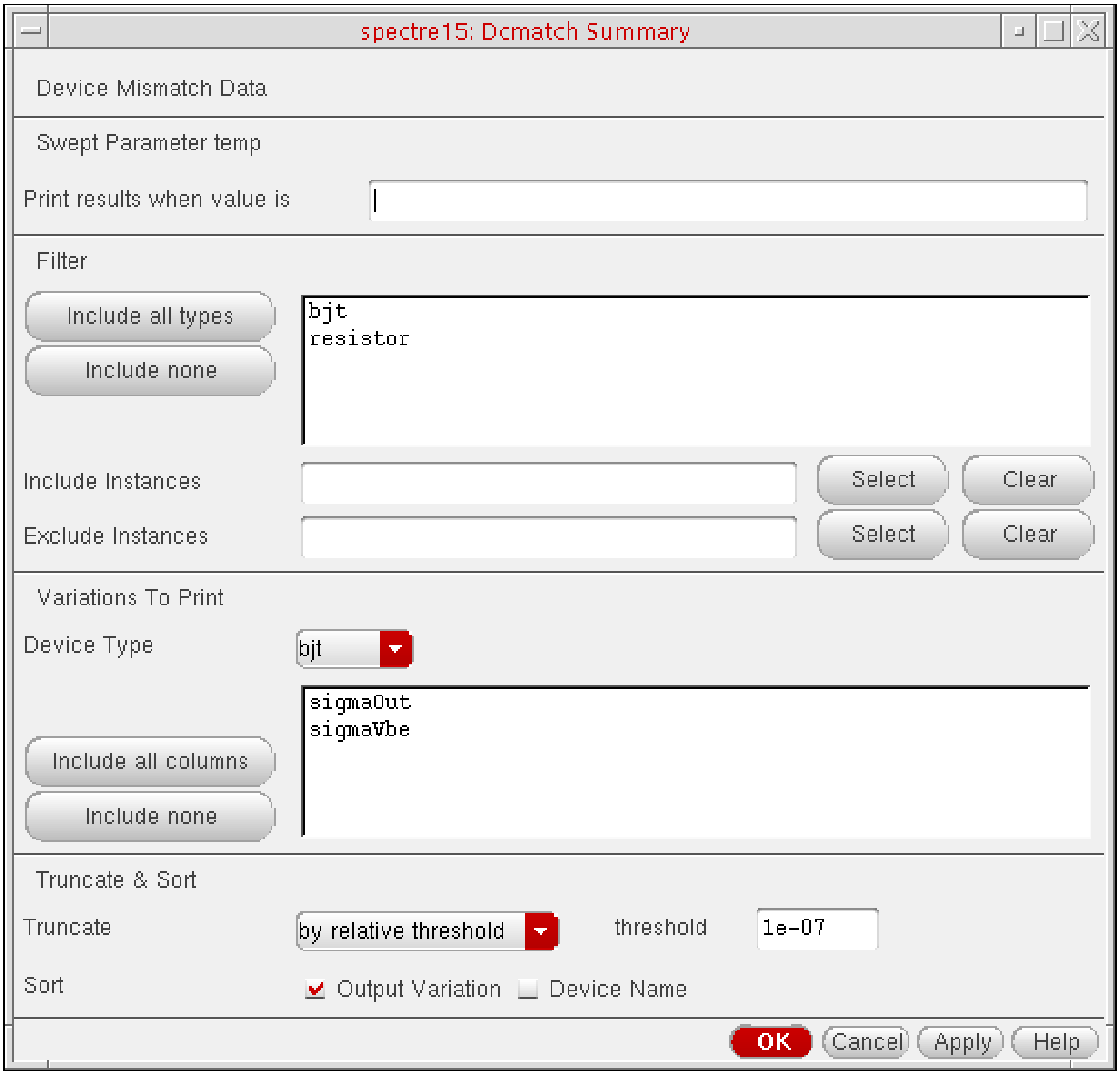
Printing DC Mismatch Summary

To print the DC mismatch summary for your circuit, do the following:

1. On the Results tab of the Outputs pane, right-click a test name and choose *Print – DC Mismatch Summary*.

   Note: This menu option is available when you run a `dcmatch` analysis or when you select a test that has `dcmatch` analysis results.

The Dcmatch Summary form appears.



2. In the *Print results when value is* field, type a value.

   Note: The *Swept Parameter* group box appears on the form only when you sweep a parameter through numeric values.

3. In the Filter group box, specify the device types you want to include in the summary.

   ❑ Click *Include all types* to include all device types.

   ❑ Click *Include none* to exclude all device types.

- ❑ Use the *Include Instances* and *Exclude Instances* fields to include and exclude specific instances.
  - ❍ Type instance names in these fields.
  - ❍ Click *Select* to select instances from the schematic.
  - ❍ Click *Clear* to clear the fields.

4. In the *Variations to Print* group box, do the following:

   a. In the *Device Type* drop-down list, select a device type.

   b. In the list area below, select one or more variations for the selected device type.
   - ❍ Click *Include all columns* to select all variations in the list area.
   - ❍ Click *Include none* to clear your selection.

5. In the *Truncate & Sort* group box, select one of the following truncation methods:

- ❑ *none* – No truncation.

- ❑ *by number* – In the *top* field, type the number of top contributors you want to see.

- ❑ *by relative threshold* – In the *threshold* field, type a relative threshold value.

- ❑ *by absolute threshold* – In the *mismatch* field, type an absolute threshold value.

6. For *Sort*, select one or both of the following sorting methods:

- ❑ *Output Variation*
- ❑ *Device Name*

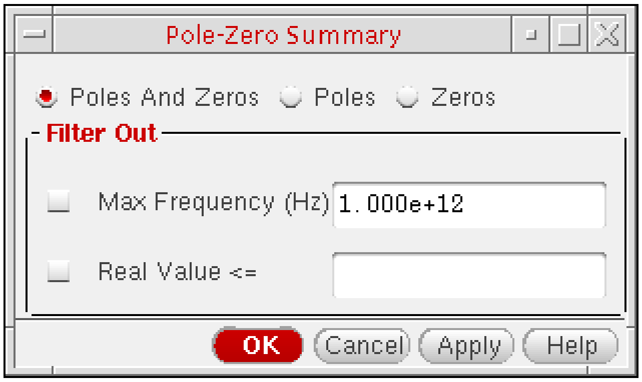
Printing Pole Zero Summary

To print the pole-zero summary for your circuit, do the following:

1. On the Results tab of the Outputs pane, right-click a test name and choose *Print – Pole-Zero Summary*.

   Note: This menu option is available when you run a pole-zero analysis or when you select a test that has pole-zero analysis results.

   The Pole-Zero Summary form appears.



2. Select one of the following options:
   - *Poles and Zeros* if you want to plot both poles and zeros.
   - *Poles* if you want to plot only poles.
   - *Zeros* if you want to plot only zeros.
3. In the *Filter Out* group box, select zero or more of the following filtering mechanisms:
   - *Max Frequency*

     This option enables you to filter out poles and zeros that are outside the frequency band of interest (FBOI) and that do not influence the transfer function in the FBOI. The default value is whatever appears in the *fmax* field on the Pole-Zero Options form. For the Direct Plot form, *fmax* is read from the header of the `psf` data. The program filters out any poles and zeros whose magnitudes exceed the frequency value you type in this field.
   - *Real Value*

     This option enables you to specify the real part of the frequency. The program filters out any poles and zeros whose real values are less than or equal to the real value you type in this field.
4. Click *OK*.

   Pole-zero data appears in the Results Display Window according to the criteria you specified.

Re-evaluating Expressions and Specifications

After running a simulation, if you add, modify or delete output expressions or specifications on the Outputs Setup tab of the Outputs pane, you can re-evaluate the expressions and specifications without having to re-run the simulation. When output expressions or specifications are re-evaluated, the updated results are displayed in the Results tab of the Outputs pane. The results database is also automatically updated.

Note: When you re-evaluate expressions and specifications, the updated results will not be plotted. For information about how to plot the results, see Plotting Results on page 624.

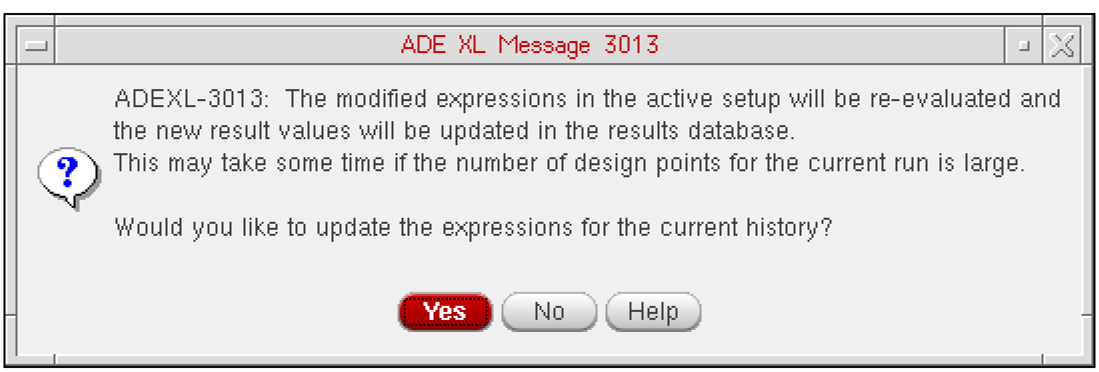
Important

If you re-evaluate expressions and specifications for an optimization run, costs are re-calculated for all the design points and a new best point is identified. If the *Save all design points* option was not selected in the Save Options form before running optimization, waveform data may not exist for the new best point.

To re-evaluate expressions and specifications for the active setup, do one of the following:

- Click the button on the Results tab of the Outputs pane to re-evaluate both expressions and specifications.
- Click the button on the Results tab of the Outputs pane, and do one of the following:
  - Select *Expressions* to re-evaluate expressions.
  - Select *Specifications* to re-evaluate specifications.
  - Select *All* to re-evaluate both expressions and specifications.

  The following message box appears.



  Click *Yes* to update the results displayed in the Results tab based on the new or modified expressions and specifications.

To re-evaluate expressions and specifications for a history item, do the following:

1. Restore the history item.

   For more information, see Restoring a Checkpoint on page 815.

2. Make the required changes in the expressions and specifications in the Outputs Setup tab of the Outputs pane.

3. Click the button on the Results tab of the Outputs pane, and do one of the following:

   ❑ Select *Expressions* to re-evaluate expressions.

   ❑ Select *Specifications* to re-evaluate specifications.

   ❑ Select *All* to re-evaluate both expressions and specifications.

See also

■ Adding an Output Expression on page 376

■ Modifying an Output Expression on page 391

■ Chapter 15, “Working with Specifications”

Saving and Restoring the Waveform Window Setup

To save and restore the waveform window setup, do the following:

1. On the Data View assistant pane, right-click the test or analysis name and choose *Save State*.

   The Saving State form appears.

2. In the *Save As* field, type a name for the state you want to save.

3. In the *What to Save* group box, select the *Waveform Setup* check box.

4. Click *OK*.

   The program saves the waveform window setup to the state name you specified.

To restore the saved settings, do the following:

1. On the Data View assistant pane, right-click the test or analysis name and choose *Load State*.

   The Loading State form appears.

2. Select a previously-saved state that contains waveform window setup information.
3. In the *What to Load* group box, select the *Waveform Setup* check box.
4. Click *OK*.

   The program loads the specified state information.

Searching for Conditional Results

Important

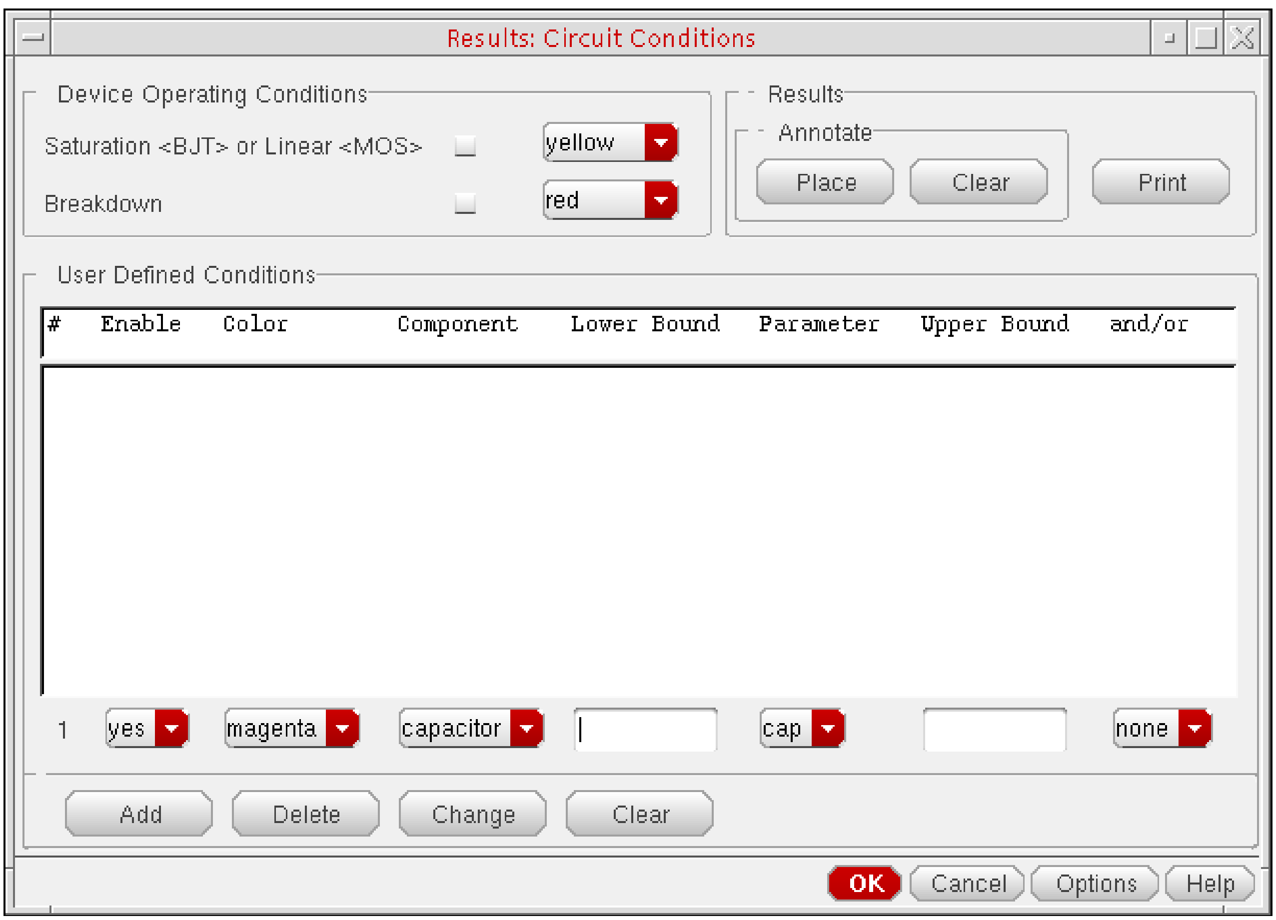
You must run a DC operating-point analysis to use the circuit conditions capability.

After running a simulation, you can search the results for components in the saturation region, breakdown region, or any user-defined region.

To do a conditional search for results, do the following:

1. On the Results tab of the Outputs pane, right-click a test name and choose *Circuit Conditions*.

   The Circuit Conditions form appears:



2. In the *Device Operating Conditions* group box, specify device operating conditions.

You can choose to view components in the saturation (for BJT devices), linear (for MOS devices), or breakdown region.

Note: The appropriate model parameters must be set for the simulator to calculate these conditions. These features might not be available for simulators other than Spectre.

3. In the *User Defined Conditions* group box, do the following:
   a. Use the cyclic and type-in fields to specify custom conditions.
   b. Click *Add*.
4. In the *Results* group box, do the following:
   - ❑ (Optional) Click *Place* to highlight the instances that meet the specified conditions on the schematic.
   - ❑ (Optional) Click *Print* to print the values of instances that meet the specified conditions in a print window.
5. Click *Add*.

See also

■ Filtering Out Components by Model Name on page 655

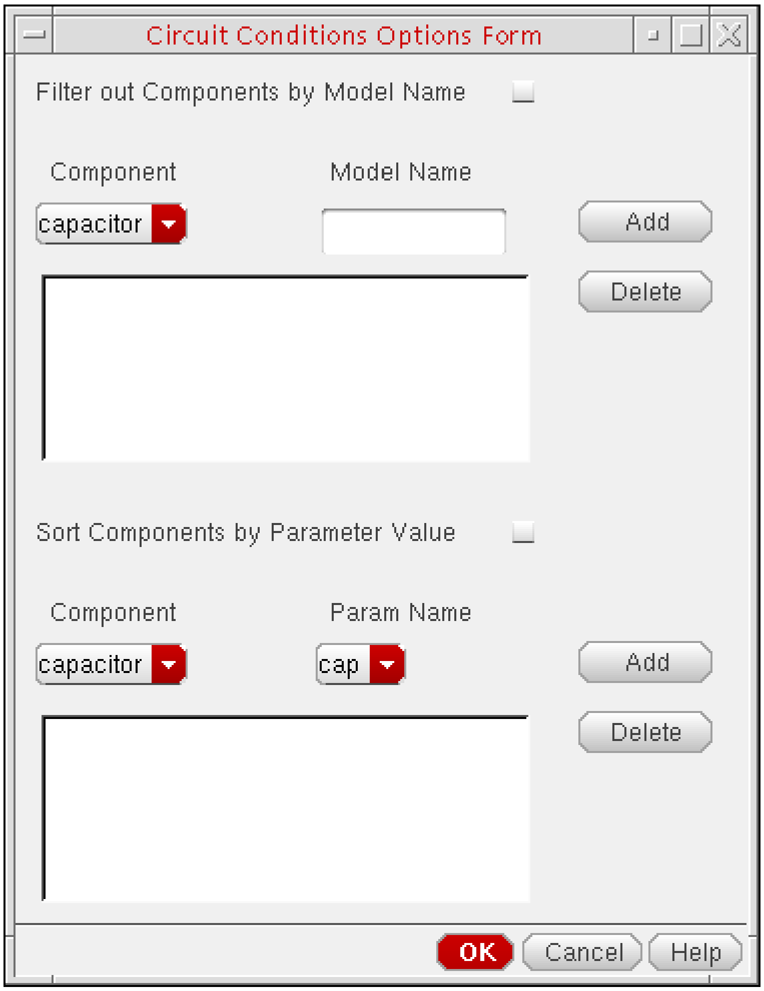
■ Sorting Components by Parameter Value on page 657

Note: Filters and sorting conditions are active only if you have selected either *and* or in the *Boolean* drop-down list at the bottom right corner of the *User Defined Conditions* group box on the Circuit Conditions form.

Filtering Out Components by Model Name

1. On the Circuit Conditions Circuit Conditions form, click *Options*.

   The Circuit Conditions Options Form appears.



2. Select the *Filter out Components by Model Name* check box.
3. In the *Component* drop-down list, select a component type.
4. In the *Model Name* field, type a model name.
5. Click *Add*.

   The specified filter appears in the list box.

Note: Filters are active only if you have selected either *and* or in the *Boolean* drop-down list at the bottom right corner of the *User Defined Conditions* group box on the Circuit

Conditions form. The program filters out any components that match the filters and these components do not appear in the output when you click *Print*.

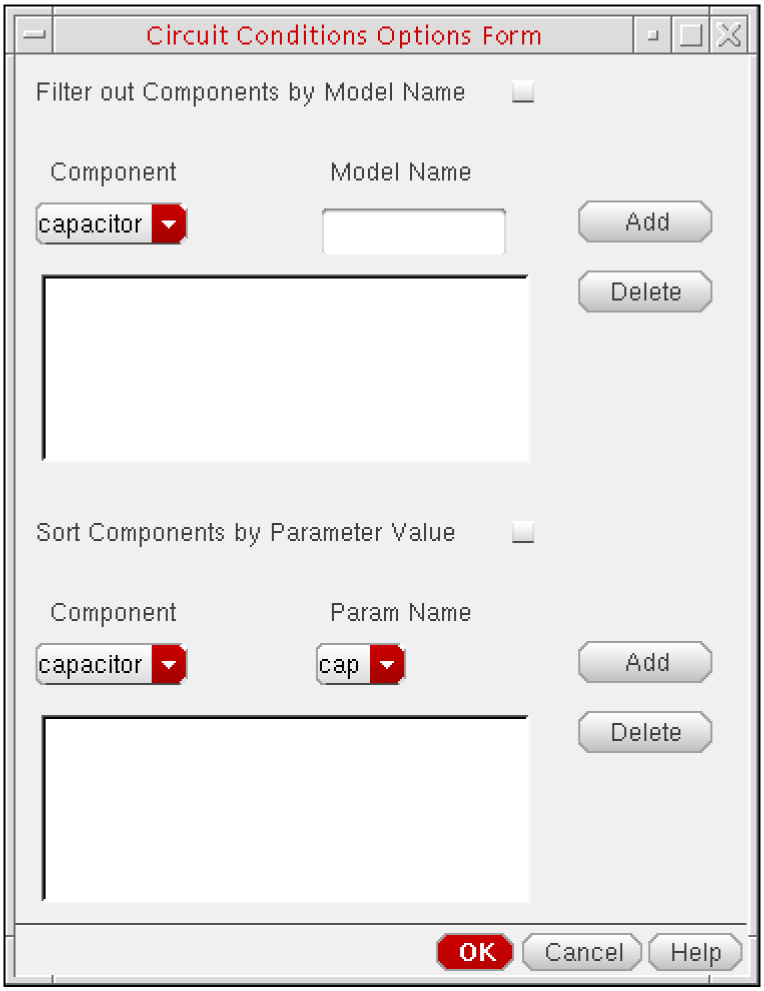
To remove a filter, do the following:

1. In the list box, select one or more filters that you want to delete.
2. Click *Delete*.

Sorting Components by Parameter Value

1. On the Circuit Conditions form, click *Options*.

   The Circuit Conditions Options Form appears.



2. Select the *Sort Components by Parameter Value* check box.
3. In the *Component* drop-down list, select a component type.
4. In the *Param Name* field, type a parameter name.
5. Click *Add*.

   The specified sort condition appears in the list box.

Note: Sort conditions are active only if you have selected either *and* or in the *Boolean* drop-down list at the bottom right corner of the *User Defined Conditions* group box on the

Circuit Conditions form. The program sorts the specified components by parameter name when you click *Print*.

Comparing Results

You can use the Spec Comparison form to compare measured values of output expressions for:

- Any two history items.
- Any two tests in the same history item or in two different history items.
- Any two design points in the same history item or in two different history items.

For more information, see the following topics:

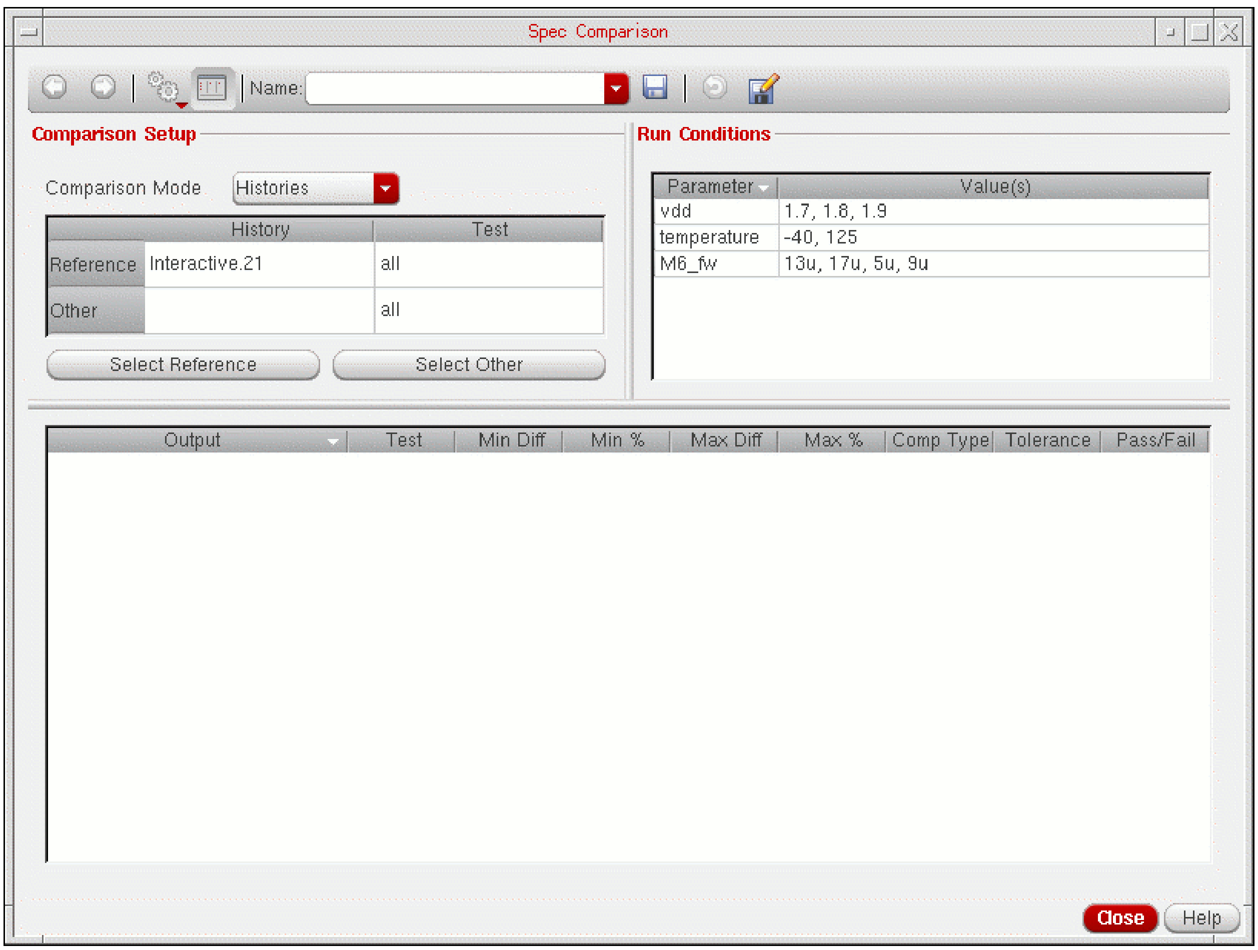
- Opening the Spec Comparison Form on page 660
- Comparing Results for History Items on page 663
- Comparing Results for Specific Tests on page 668
- Comparing Results for Specific Design Points on page 670
- Comparing the Detailed Results for Output Expressions on page 675
- Hiding and Showing the Comparison Data for Tests on page 678
- Hiding and Showing an Output Expression in the Spec Comparison on page 678
- Updating the Spec Comparison with the Latest Results on page 678
- Sorting Data in the Spec Comparison Form on page 678
- Saving a Spec Comparison on page 679
- Opening a Spec Comparison on page 679
- Deleting a Spec Comparison on page 679
- Exporting a Spec Comparison to a HTML or CSV File on page 679

Opening the Spec Comparison Form

To open the Spec Comparison form, do one the following:

- Choose *Create – Spec Comparison*.
- Click the  toolbar button on the ADE XL window.
- Click the  button on the Results tab.

  The Spec Comparison form appears.



The toolbar in the Spec Comparison form is described in the following table:

Table 14-8 Spec Comparison Toolbar

Icon	Name	Description
	Back to Summary View	Switches from the detail view to the summary view. For more information, see Comparing the Detailed Results for Output Expressions on page 675.
	Show Detail View	Displays the comparison of detailed results for the selected output expression in the detail view. For more information, see Comparing the Detailed Results for Output Expressions on page 675.
	Select which tests are displayed in the table	Hides or shows the comparison data for specific tests. For more information, see Hiding and Showing the Comparison Data for Tests on page 678.
	Show/Hide Setup Conditions Tables	Hides or shows the *Comparison Setup* and *Run Conditions* group boxes in the Spec Comparison form.
Name	Name of Saved Spec Comparison	Lets you do the following: ■ Specify the name by which a spec comparison is to be saved. For more information, see Saving a Spec Comparison on page 679. ■ Open an existing spec comparison. For more information, see Opening a Spec Comparison on page 679
	Save Spec Comparison	Saves the spec comparison with the name specified in the *Name* field. For more information, see Saving a Spec Comparison on page 679.
	Update with Latest Results Data	Updates the spec comparison with the latest results from the history items selected for comparison. For more information, see Updating the Spec Comparison with the Latest Results on page 678.

Icon	Name	Description
	Export to CSV or HTML File	Exports a spec comparison to a HTML or comma-separated values (CSV) file. For more information, see <u>Exporting a Spec Comparison to a HTML or CSV File</u> on page 679.

The fields in the Spec Comparison form are described in the following table:

Field	Description
Comparison Setup	
Comparison Mode	Specifies the comparison mode. Do one of the following: ■ Select *Histories* to compare measured values of output expressions for: ❑ Any two history items. For more information, see <u>Comparing Results for History Items</u> on page 663. ❑ Any two tests in the same history item or in two different history items. For more information, see <u>Comparing Results for Specific Tests</u> on page 668. ■ Select *Design Points* to compare measured values of output expressions for any two design points in the same history item or in two different history items. You can compare two design points in the same test or in two different tests. For more information, see <u>Comparing Results for Specific Design Points</u> on page 670.
History	Displays the history items for which you are comparing the measured values of output expressions. For more information, see <u>Comparing Results for History Items</u> on page 663.

Field	Description
Test	Displays the tests in the history items for which you are comparing the results. For more information, see Comparing Results for Specific Tests on page 668.
Point	Displays the design points in the history items for which you are comparing the measured values of output expressions. This column is displayed only if the *Design Points* is selected in the Comparison Mode drop-down list. For more information, see Comparing Results for Specific Design Points on page 670.
Run Conditions	Displays the conditions that were used for the simulation run in the reference history item. The values of each swept parameter in the reference history item is displayed as a list or range of values. For example, if a parameter `p` is swept through a list of values `x`, `y`, and `z`, then the *Value(s)* column displays the values as `p=x,y,z` If a parameter `p` is swept through a range of values, the *Value(s)* column displays the values as `p=startValue:increment:stopValue`
Parameter	Displays the names of swept parameters.
Value(s)	Displays the values of swept parameters.

Comparing Results for History Items

To compare results for history items, do the following:

1. From the *Comparison Mode* drop-down list, choose *Histories*.
2. The history item for which you are viewing the results in the Results tab is displayed as the default reference history item. Do one of the following to select a different reference history item:
   - ❑ Double-click in the *History* column in the *Reference* row and select a different history item from the drop-down list.

- ❑ Click the *Select Reference* button, click the tab for the history item in the Results tab, then click anywhere on the results displayed for the history item.

Note: The Spec Comparison form does not allow comparison of results of history items for optimization runs. So you cannot select history items for optimization runs when the comparison mode is *Histories*.

3. Do one of the following to select the history item whose results you want to compare with the reference history item.

- ❑ Double-click in the *History* column in the *Other* row and select the history item from the drop-down list.
- ❑ Click the *Select Other* button, click the tab for the history item in the Results tab, then click anywhere on the results displayed for the history item.

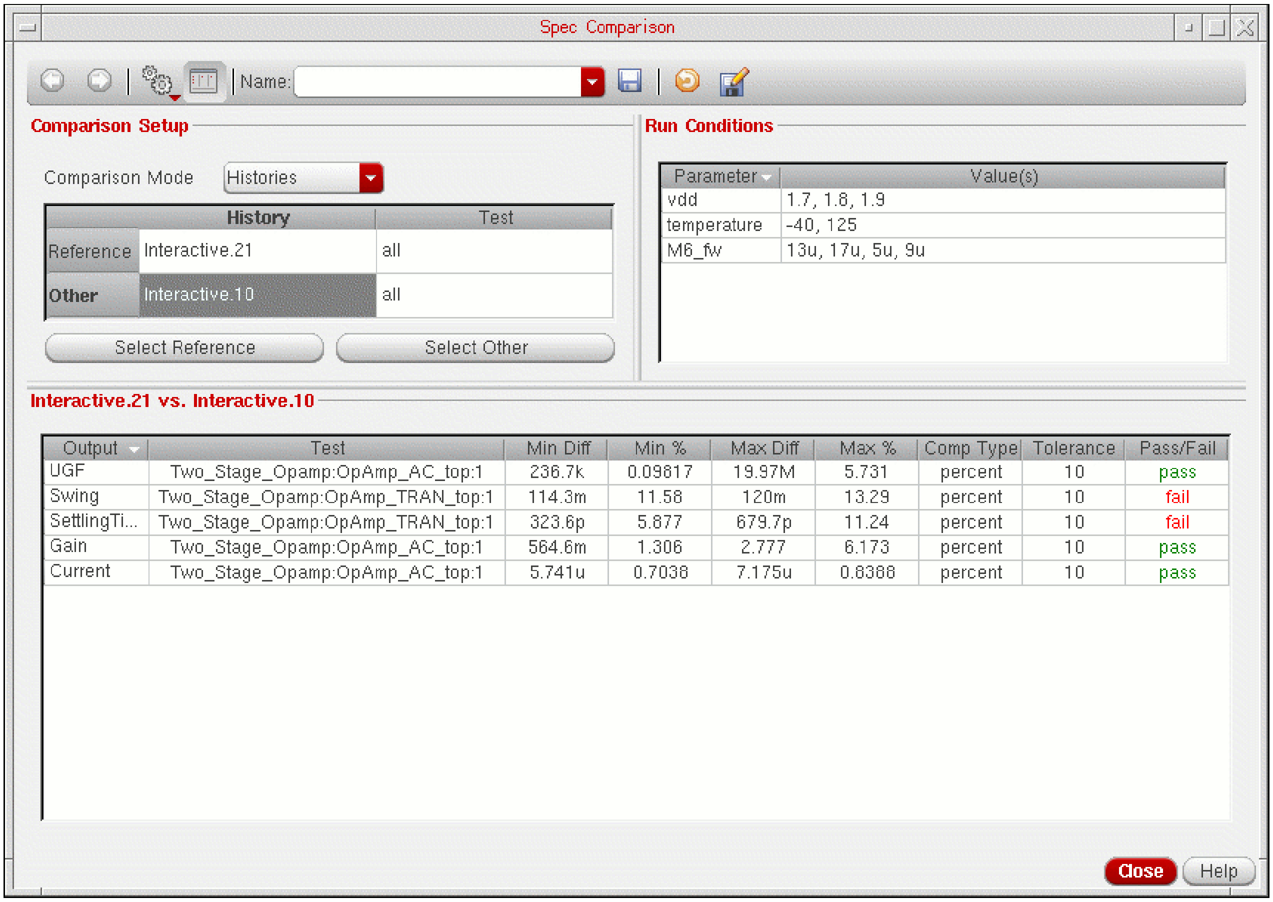
Note: Earlier, in the *Other* row, you could select only that history item that has the same set of conditions (displayed in the *Run Conditions* field) as the reference history item. Starting IC 6.1.5, you can select a history with different run conditions, for example, different sweep values for variables.

Tip

After you click *Select Other*, you can select any number of history items in the Results tab without clicking the button again. For example, you can click the *Select Other* button and select a history item in the Results tab to view the results comparison. Once you are done, you can select another history item in the Results tab without clicking *Select Other* again.

The comparison of results for the history items is displayed in the Spec Comparison form.

Figure 14-20 Spec Comparison for History Items



Note: By default, the Spec Comparison form displays the comparison of results for all the tests in the reference and other history item. The text *all* in the *Test* column indicates that the results for all the tests will be compared. For information about viewing the spec comparison for specific tests, see Comparing Results for Specific Tests on page 668.

The columns that are displayed in the Spec Comparison form when you compare the specifications for two history items are described in the following table.

Table 14-9 Spec Comparison Columns for History Items and Tests

Column	Description
Output	Displays the names of output expressions for which the results are being compared. **Note:** Output expressions that evaluate to waveforms are not displayed in the Spec Comparison form.
Test	Lists the tests used to generate the results for the output expressions.
Min Diff	Displays the minimum difference between the measured values across all the points in the two history items.
Min %	Displays the percentage error between the two measured values in the history items for which the minimum difference is reported.
Max Diff	Displays the maximum difference between the measured values across all the points in the two history items.
Max %	Displays the percentage error between the two measured values in the history items for which the maximum difference is reported.

<table>
<tr><th>Column</th><th>Description</th></tr>
<tr><td>Comp Type</td><td>Comparison type, selected from a drop-down list, that specifies how the results from the two history items are compared. Pass/fail criteria are as follows:

absolute—the absolute value of the difference in measured values for the two history items must be less than the tolerance specified. If absolute, the output expression will have a pass status if:
<code>abs(refHistoryValue - otherHistoryValue) < Tolerance</code>
percentage—the absolute value of the difference in measured values for the two history items must be less than the tolerance multiplied by the absolute value of the measured value for the reference history item. If percentage, the output expression will have a pass status if:
<code>abs(refHistoryValue - otherHistoryValue) <
abs(refHistoryValue * Tolerance)</code>
range—the absolute value of the difference in measured values for the two history items must be less than the tolerance multiplied by the absolute value of the difference between the maximum and minimum measured values for the reference history item. If range, the output expression will have a pass status if:
<code>abs(refHistoryValue - otherHistoryValue) <
abs(maxrefHistoryValue - minOtherHistoryValue) *
Tolerance</code>
absolute/percentage—the absolute value of the difference in measured values for the two history items must be less than the specified tolerance, or less than the tolerance multiplied by the absolute value of the measured value for the reference history item (see equations for absolute and percentage, above). This comparison type is useful for comparing measured values near zero.
none—does not apply the tolerance to determine pass/fail criteria. The pass/fail status will also not be displayed in the Pass/Fail column.
</td></tr>
</table>

Column	Description
Tolerance	Specifies the tolerance value that determines the pass or fail status of the output expression. **Note:** Specify a positive value if the comparison type is *absolute* or *range*, and a positive percentage if the comparison type is *percentage*.
Pass/Fail	Displays the pass or fail status of the output expression based on the specified comparison type and the tolerance.

See also:

- Comparing the Detailed Results for Output Expressions on page 675
- Hiding and Showing the Comparison Data for Tests on page 678
- Hiding and Showing an Output Expression in the Spec Comparison on page 678
- Updating the Spec Comparison with the Latest Results on page 678
- Sorting Data in the Spec Comparison Form on page 678
- Exporting a Spec Comparison to a HTML or CSV File on page 679

Comparing Results for Specific Tests

By default, the Spec Comparison form displays the comparison of measured values of output expressions for all the tests in the reference and other history item. The text *all* in the *Test* column indicates that the results for all the tests will be compared.

You can compare the results for any two tests in the same history item or in two different history items.

To compare the results for specific tests, do the following:

1. Select the reference and other history item by performing steps 1 to 3 described in Comparing Results for History Items on page 663.

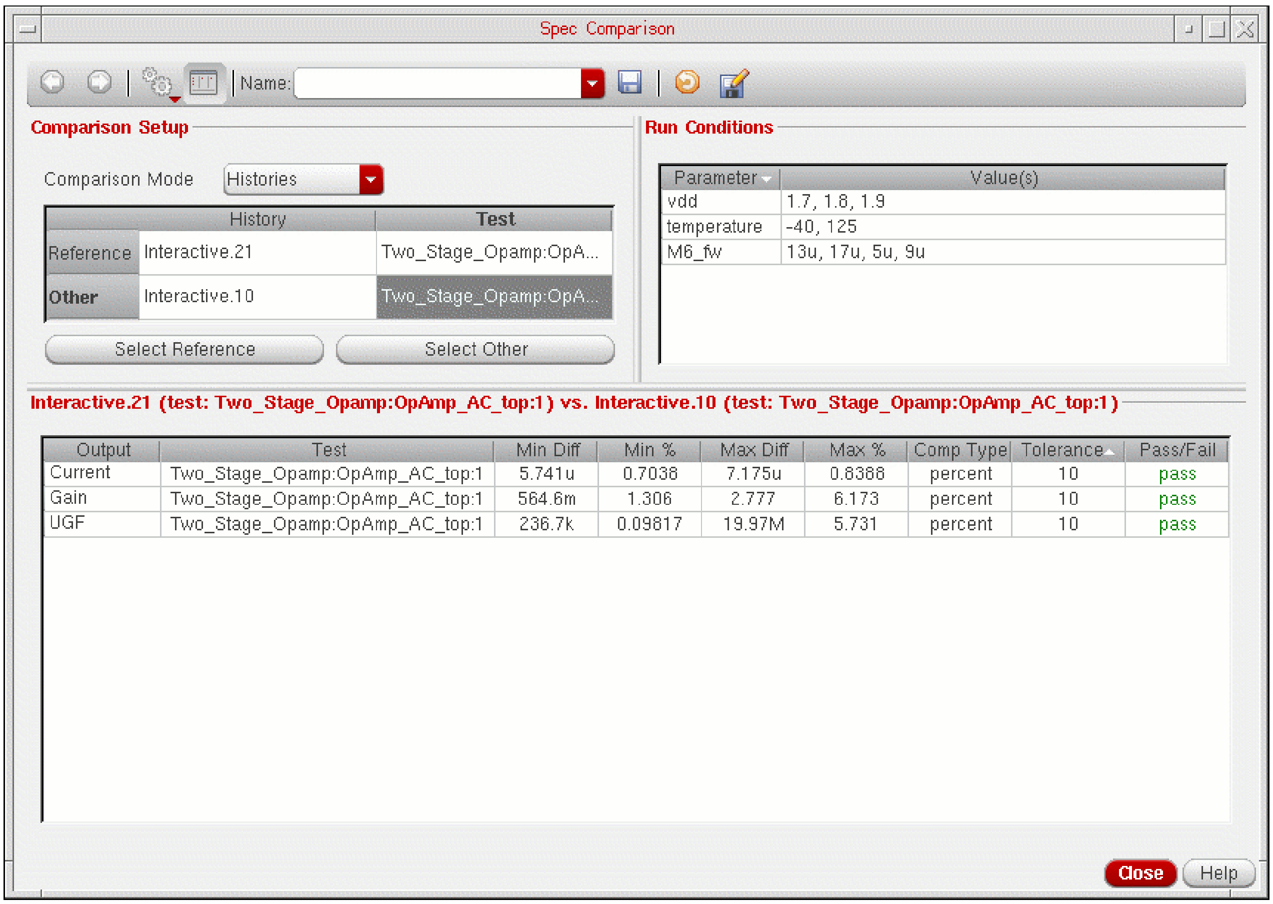
   Note: To compare the results for any two tests in the same history item, select the same history item as the reference and other history item.

2. Double-click in the *Test* column in the *Reference* row and select a test name for the reference history item from the drop-down list.

3. Double-click in the *Test* column in the *Other* row and select a test name for the other history item from the drop-down list.

The comparison of results for the tests is displayed in the Spec Comparison form.

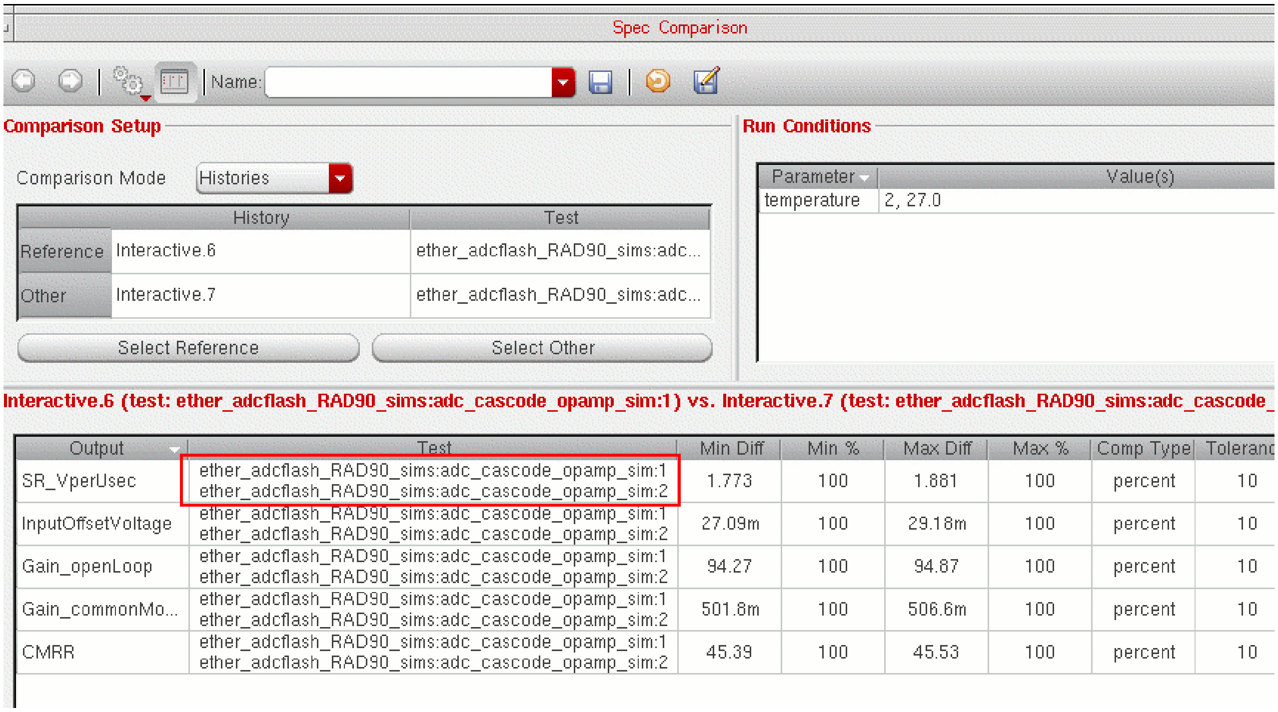
Figure 14-21 Spec Comparison for Tests



The columns that are displayed in the Spec Comparison form when you compare the specifications for two tests are described in Table 14-9 on page 666.

When you compare results of different tests in two history items, the *Test* column shows the names of both the referenced test and the other test, as shown below.

Figure 14-22 Spec Comparison for Different Tests



See also:

- Comparing the Detailed Results for Output Expressions on page 675
- Hiding and Showing an Output Expression in the Spec Comparison on page 678
- Updating the Spec Comparison with the Latest Results on page 678
- Sorting Data in the Spec Comparison Form on page 678
- Exporting a Spec Comparison to a HTML or CSV File on page 679

Comparing Results for Specific Design Points

By default, the Spec Comparison form displays the comparison of measured values of output expressions for all the design points in the reference and other history item.

You can compare the results for any two design points in the same history item or in two different history items.

To compare the results for specific design points, do the following:

1. From the *Comparison Mode* drop-down list, choose *Design Points*.

2. Select the reference and other history item by performing steps 2 and 3 described in Comparing Results for History Items on page 663.

3. By default, the Spec Comparison form displays the comparison of results for design points in all the tests in the reference and other history item. To compare the results for design points in specific tests, select the tests by performing steps 2 and 3 described in Comparing Results for Specific Tests on page 668.

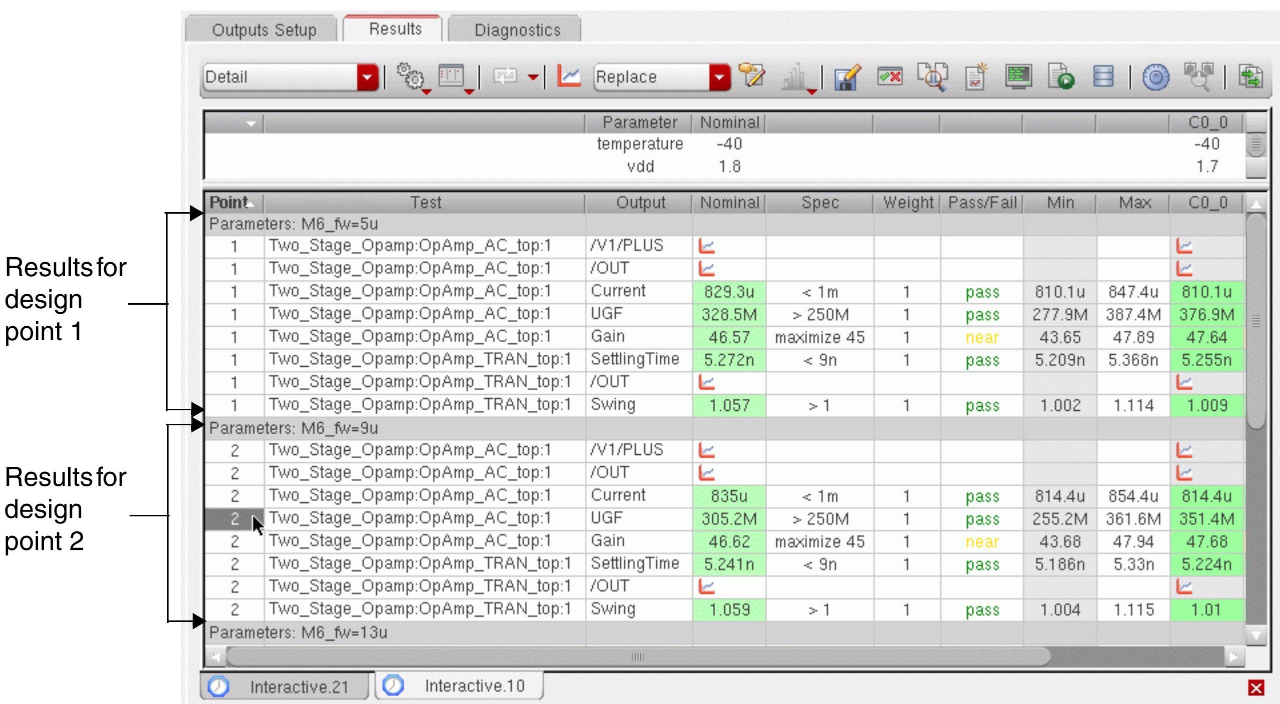
4. By default, the ID of the first design point is displayed as the default design point for the reference history item. If the reference history item is for an optimization run, the ID of the best design point in the reference history item is displayed as the default point for the reference history item.

   Do one of the following to select a different design point for the reference history item:

   ❑ Double-click in the *Point* column in the *Reference* row and specify a different design point.

   ❑ Click on the *Select Reference* button and click on the results for a specific design point for the history item in the Results tab.

   The following figure shows an example of selecting design point `2` for comparing results by clicking on the results for design point `2`.



5. Do one of the following to select a design point for the other history item:

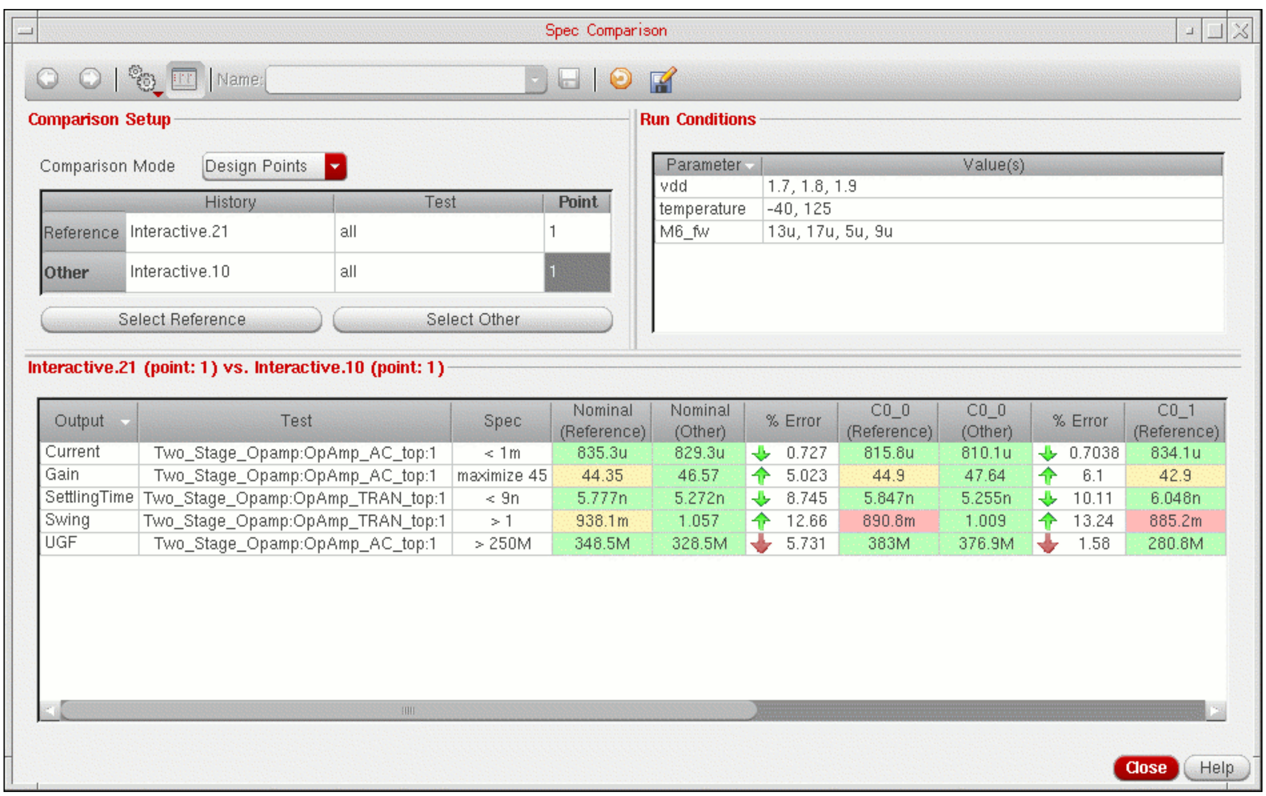
- ❑ Double-click in the *Point* column in the *Other* row and select a design point for the other history item from the drop-down list.
- ❑ Click on the *Select Other* button and click on the results for a specific design point for the history item in the Results tab.

Tip

After you click *Select Other* , you can select any number of design points in the Results tab without clicking the button again. For example, you can click the *Select Other* button, and select a design point in the Results tab to view the results comparison. Once you are done, you can select another design point in the Results tab without clicking *Select Other* again.

The comparison of results for the design points is displayed in the Spec Comparison form.

Figure 14-23 Spec Comparison for Design Points



The columns that are displayed in the Spec Comparison form when you compare the results for two design points are described in the following table.

Table 14-10 Spec Comparison Columns for Design Points

Column	Description
Output	Displays the names of output expressions for which the results are being compared. **Note:** Output expressions that evaluate to waveforms are not displayed in the Spec Comparison form.
Test	Lists the tests used to generate the results for output expressions.
Spec	Displays specification information.
Nominal (Reference)	Displays the status and measured value for expressions at the nominal corner in the reference history item. The color of a cell indicates the status of the result with respect to the specification: ■ Green indicates a value within the specifications. ■ Yellow indicates a value that is no more than 10% outside the target value of the specification. ■ Red indicates a value that is more than 10% outside the target value of the specification.
Nominal (Other)	Displays the status and measured value for expressions at the nominal corner in the other history item.

Column	Description
% Error	Displays the status (using arrow icons) and percentage error of the measured values at the nominal corner in the other history item relative to the values in the reference history item. The arrow color indicates how well the specification is satisfied by the value in the other history item relative to the value in the reference history item. A green arrow indicates that the value in the other history item more effectively meets the specification relative to the reference history item, whereas a red arrow indicates that the value in the other history item is less effective at meeting the specification. The arrow direction indicates the magnitude of the value in the other history item relative to the value in the reference history item. An up arrow indicates that the value in the other history item is higher relative to the reference history item, whereas a down arrow indicates that the value in the other history item is lower. For example, the  icon indicates that the value in the other history item more effectively meets the specification and has a higher value relative to the value in the reference history item.
cornerName (Reference)	Displays the status and measured value for expressions at each corner in the reference history item. The color of a cell indicates the status of the result with respect to the specification: ■ Green indicates a value within the specifications. ■ Yellow indicates a value that is no more than 10% outside the target value of the specification. ■ Red indicates a value that is more than 10% outside the target value of the specification.
cornerName (Other)	Displays the status and measured value for expressions at each corner in the other history item.

Column	Description
% Error	Displays the status (using arrow icons) and percentage error of the measured values at the corner in the other history item relative to the values in the reference history item. The arrow color indicates how well the specification is satisfied by the value in the other history item relative to the value in the reference history item. A green arrow indicates that the value in the other history item more effectively meets the specification relative to the reference history item, whereas a red arrow indicates that the value in the other history item is less effective at meeting the specification. The arrow direction indicates the magnitude of the value in the other history item relative to the value in the reference history item. An up arrow indicates that the value in the other history item is higher relative to the reference history item, whereas a down arrow indicates that the value in the other history item is lower. For example, the icon indicates that the value in the other history item more effectively meets the specification and has a higher value relative to the value in the reference history item.

See also:

- Comparing the Detailed Results for Output Expressions on page 675
- Hiding and Showing an Output Expression in the Spec Comparison on page 678
- Updating the Spec Comparison with the Latest Results on page 678
- Sorting Data in the Spec Comparison Form on page 678
- Exporting a Spec Comparison to a HTML or CSV File on page 679

Comparing the Detailed Results for Output Expressions

You can compare the detailed results for output expressions when the *Histories* option is selected in the *Comparison Mode* drop-down list.

To compare the detailed results for an output expression, do one of the following:

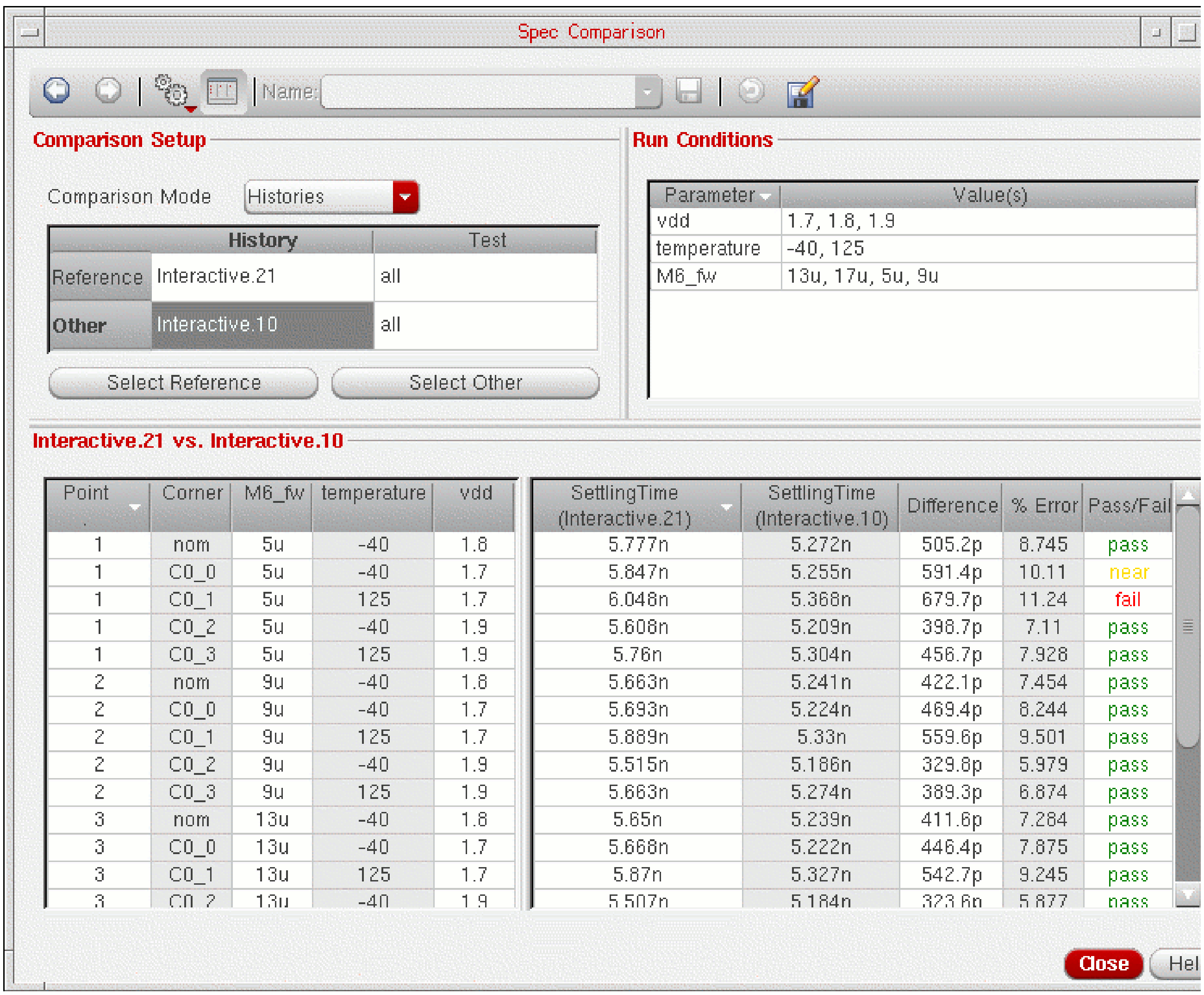
- Select the row for the output expression and click the button.

- Double-click on the row for the output expression.
- Right-click on the row for the output expression and choose *Show Detail View*.

The comparison of the detailed results are displayed in the detail view.

Note: You cannot compare the detailed results for an output expression if the reference or other history item does not exist.

Figure 14-24 Spec Comparison Detailed View



The columns in the detail view in the Spec Comparison form are described in the following tables.

Table 14-11 Columns in Table on Left Side of Spec Comparison Detailed View

Column	Description
Point	Displays the design point number.
Corner	Displays the name of the corner.
paramName	Displays the values of swept parameters at each design point.

Table 14-12 Columns in Table on Right Side of Spec Comparison Detailed View

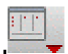
Column	Description
exprName (*referenceHistoryName*)	Displays the measured value for the expression in the reference history item at each design point.
exprName (*otherHistoryName*)	Displays the measured value for the expression in the other history item at each design point.
Difference	Displays the difference between the measured values for the expression in the reference history item and the other history item.
% Error	Displays the percentage error of the measured values for the expression in the other history item relative to the values in the reference history item.
Pass/Fail	Displays a *pass*, *near* or *fail* status for the specification. ■ *pass* means that all the measured values are within the limits defined by the specification. ■ *near* means that the measured values are no more than 10% outside the target value of the specification. ■ *fail* means that the measured values are greater than 10% outside the target value of the specification.

To go back to the summary view, do one of the following in the detail view:

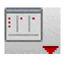
- Click the  button.
- Right-click on any row and choose *Go back*.

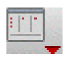
Hiding and Showing the Comparison Data for Tests

By default, the Spec Comparison form displays the comparison data for all the tests in the reference and other history item.

Click the  button. A check mark next to a test indicates that data for that test is being displayed.

To hide the comparison data for a test, do the following:

➤ Click the  button and choose the test.

To unhide the data click the  button once again and choose the test.

Hiding and Showing an Output Expression in the Spec Comparison

To hide an output expression in the Spec Comparison form, do the following.

➤ Right-click on the row for the output expression and choose *Hide Selected Rows*.

To hide the rows for multiple output expressions, hold down the *Shift* key (for contiguous selection) or the *Ctrl* key (for noncontiguous selection), right-click and choose *Hide Selected Rows*.

To display all the output expressions in the Spec Comparison form, do the following:

➤ Right-click on any row and choose *Show All Hidden Rows*.

Updating the Spec Comparison with the Latest Results

To update the spec comparison with the latest results from the history items selected for comparison, do the following:

➤ Click the  button.

Note: You cannot update the spec comparison with the latest results if the reference or other history item does not exist.

Sorting Data in the Spec Comparison Form

To sort data in the Spec Comparison form, do the following:

➤ Click on the name of the column based on which you want to sort the data.

Saving a Spec Comparison

To save a spec comparison, do the following:

1. In the *Name* field, type a name for the spec comparison.
2. Click the  button.

The spec comparison is saved to the `documents` folder of the ADE XL view and displayed in the *Documents* tree on the Data View pane. For more information about working with documents, see Chapter 18, "Working with Documents."

Note: The spec comparison file is saved with the `.speccomparison` extension.

Opening a Spec Comparison

To open an existing spec comparison, do one of the following:

- In the *Name* cyclic field on the Spec Comparison form, select the spec comparison.
- In the *Documents* tree on the Data View pane, double-click on the spec comparison.

The spec comparison is displayed in the Spec Comparison form.

Deleting a Spec Comparison

To delete an existing spec comparison, do the following:

➡ In the *Documents* tree on the Data View pane, right-click the spec comparison and choose *Delete*.

The spec comparison will not be displayed in the *Name* drop-down list in the Spec Comparison form.

Exporting a Spec Comparison to a HTML or CSV File

You can export the spec comparison data displayed in the Spec Comparison form to a HTML or comma-separated values (CSV) file.

Tip

You can hide the spec comparison data you do not want to export to the HTML or CSV file. For more information on hiding spec comparison data, see the following topics:

- Hiding and Showing the Comparison Data for Tests on page 678
- Hiding and Showing an Output Expression in the Spec Comparison on page 678

To export a spec comparison to a HTML or CSV file, do the following:

1. Click the  button.

   The Export Results form appears.

2. In the *File name* field, type a file name with the `.html`, `.htm` or `.csv` extension.
3. Click *Save*.

   The program exports the results to the file you specified. By default, the file is saved in the `documents` folder of the ADE XL view and displayed in the *Documents* tree on the Data View pane. For more information about working with documents, see Chapter 18, "Working with Documents."

Note: If you click the  button in the detail view, the resulting HTML or CSV file contains the contents of the detail view and summary view.

Saving Results

To save results for specific tests, do the following:

1. On the Results tab of the Outputs pane, right-click a test name and choose *Save*.

   The Save Results form appears.

2. (Optional) By default, the results are saved to a directory named `schematic-save`. To use a different directory name, type the directory name in the *Save As* field.

3. (Optional) In the *Comment* field, type a comment so that you can more easily differentiate simulation results.

4. Enter the path to the directory where you want to save the results.

   Alternatively click the browse button to specify the directory.

5. Click *OK*.

To save results for all tests, do the following:

1. Click the  button on the Results tab of the Outputs pane.

   The Save Results form appears.

2. Enter the path to the directory where you want to save the results.

   Alternatively click the browse button to specify the directory.

3. (Optional) Select the *Copy PSF Results?* check box if you want to copy PSF results.

4. Click *OK*.

Working in the Results Display Window

The Results Display Window appears automatically when you right-click a test name on the Results tab of the Outputs pane and choose an item from the *Print* submenu.

Using the menus in the Results Display Window, you can perform the following tasks:

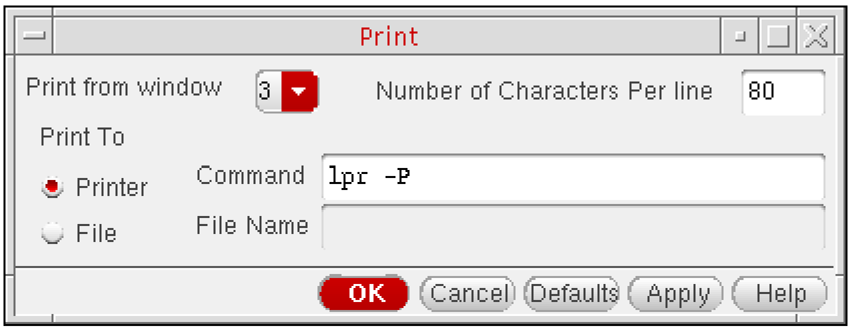
- Printing Results from the Results Display Window on page 683
- Saving Results Display Window Setup Information on page 684
- Loading Results Display Window Setup Information on page 684
- Updating Results in the Results Display Window on page 685
- Clearing the Results Display Window on page 685
- Making a Window Active on page 685
- Closing a Results Display Window on page 685
- Editing Expressions on page 686
- Setting Results Display Options on page 688
- Displaying Untruncated Output Information on page 690

Printing Results from the Results Display Window

To print results from the Results Display Window, do the following:

1. Choose *Window – Print*.

   The Print form appears.



2. In the *Print from window* drop-down list, verify that the number from the lower left corner of the Results Display Window appears.
3. In the *Number of Characters Per Line* field, type an integer number of characters you want printed on each line.
4. Select one of the following *Print To* options:
   - ❑ *Printer* – In the *Command* field, type a valid printer command.
   - ❑ *File* – In the *File Name* field, type a file name.
5. Click *OK*.

   The program prints the contents of your Results Display Window to a file or printer.

Saving Results Display Window Setup Information

You can save your current Results Display Window setup (such as printing format, setting a printing range if the amount of data is too large, printing at a certain interval, sorting) to a file so that you can load it again after running another simulation. The saved setup information determines how the program displays your results.

Note: You can only save setup information for expressions that can evaluate to waveforms: If you print a single number, like a node voltage, the *Save State* command is not active.

To save the contents and format of a Results Display Window, do the following:

1. Choose *Window – Save State*.

   The Save Window form appears.

2. In the empty field, type a file name.
3. Click *OK*.

   The program saves the setup of your current Results Display Window.

Loading Results Display Window Setup Information

To load a Results Display Window state that you previously saved, do the following:

1. Choose *Window – Load State*.

   The Load Window form appears.

2. In the empty field, type the name of the saved window state file.
3. Click *OK*.

The program loads the specified window state file.

Updating Results in the Results Display Window

To update the Results Display Window with results from a new simulation, do the following:

➤ Choose *Window – Update Results*.

The program uses the current window setup when updating the results data. The program updates only results data that can evaluate to a waveform: This command is not active for updating single-number data (such as a node voltage).

Clearing the Results Display Window

To clear the results from the Results Display Window, do the following:

➤ Choose *Window – Clear*.

The program clears the Results Display Window.

Making a Window Active

There is no limit to the number of Results Display Windows you can have open, but only one is active at a time. The program writes printed results to the active window.

To make a window active, do the following:

➤ Choose *Window – Make Active*.

The word *Active* appears beneath the menu bar in the Results Display Window.

Closing a Results Display Window

To close a Results Display Window, do the following:

➤ Choose *Window – Close*.

The Results Display Window closes.

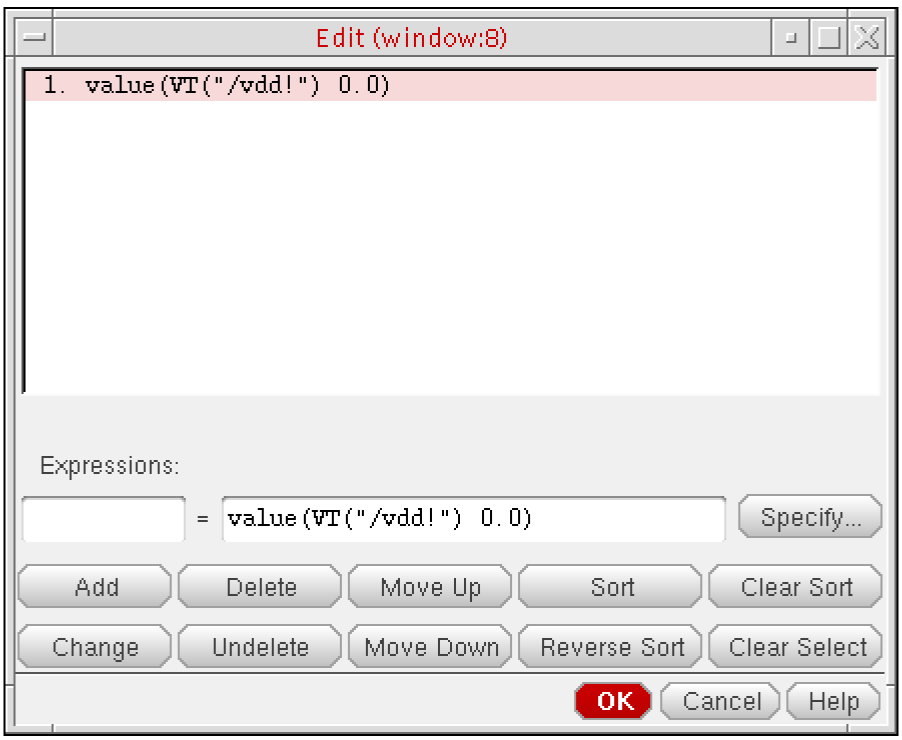
Editing Expressions

You can edit any expressions that evaluate to waveforms (such as DC operating point, transient operating point, and model parameters). If you print only one value, the *Edit* menu choices are not active. The editing commands operate only on the last table in the active Results Display Window.

To edit expressions in the currently active Results Display Window, do the following:

1. Choose *Expressions – Edit*.

   The Edit form appears.



2. Using the buttons and fields on the form, edit the expressions you want to edit. When more than one expression appears in the window, you must select the expression you intend to edit.

Expressions	If you specified an optional aliased name for the expression, that name appears in the edit field to the left of the equal sign. Otherwise, that field is blank. The expression appears in the edit field to the right of the equal sign.

Specify	Retrieves the expression from the calculator buffer into the edit field. **Note:** If you click *Specify* when there is no expression in the buffer, the Calculator window appears so that you can create an expression.
Add	Adds the expression from the edit field to the list box and as a column in your Results Display Window. If you specified an optional name, this name appears in the list box and as the column heading.
Change	Replaces the selected expression in the list box with the one in the edit field and its values in the Results Display Window.
Delete	Removes the selected expression from the list box and its column from the Results Display Window.
Undelete	Undoes the last *Delete* action.
Move Up	Moves the selected expression up one position in the list box and its column over one to the left in the Results Display Window. If the selected expression is already at the top of the list, it moves (wraps around) to the bottom and its column moves (wraps around) to the rightmost side.
Move Down	Moves the selected expression down one position in the list box and its column over one to the right in the Results Display Window. If the selected expression is already at the bottom of the list, it moves (wraps around) to the top and its column moves (wraps around) to the leftmost side.
Sort	Sorts the values of the selected expression (which appear in the Results Display Window) such that they increase down the column (smallest value on top).
Reverse Sort	Sorts the values of the selected expression (which appear in the Results Display Window) such that they decrease down the column (largest value on top).
Clear Sort	Reverts to the default sort order (alphabetical by parameter name).
Clear Selection	Clears the selection in the list box. **Note:** You can also clear a selection by clicking it while holding down the *Control* key.

3. Click *OK* when you are finished editing.

Note: See also "Saving Results Display Window Setup Information" on page 684.

Setting Results Display Options

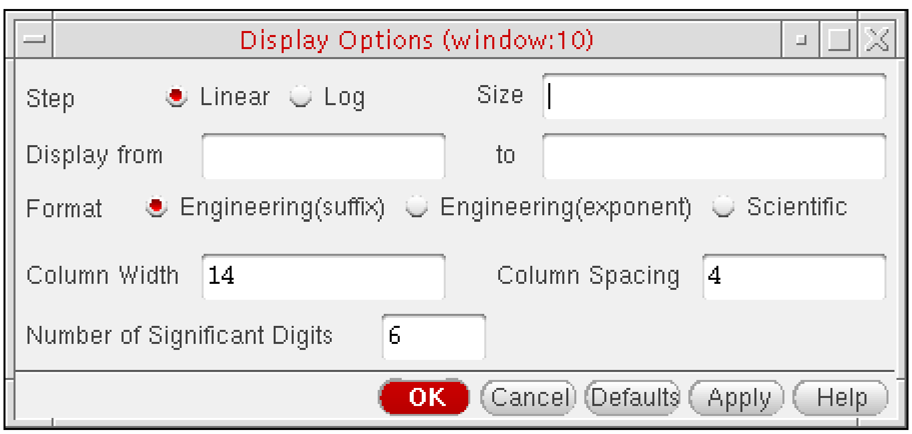
Tip

The Results Display Window may contain more than one type of results. Display options settings apply only to the last result (if the last result can evaluate to a waveform). After you edit the data, only the last results appear in the window. You can preserve the previous results by opening a new Results Display Window and printing the results you want to edit in the new window.

To change the results display options in the currently active Results Display Window, do the following:

1. Choose *Expressions – Display Options*.

   The Display Options form appears.



2. Use the fields and buttons on the form to specify one or more of the following display options:

Step — Specifies one of the following scales for the step size:

- *Linear*
- *Log*

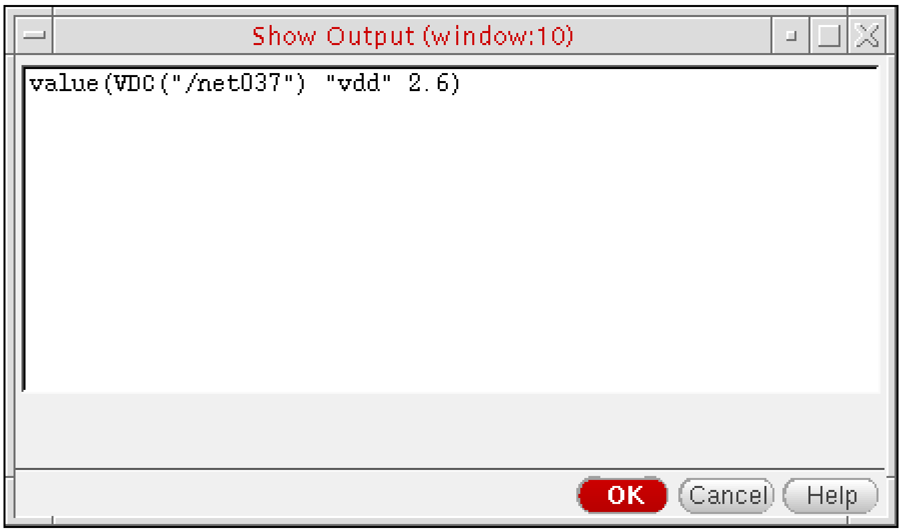
Size	Specifies the interval for printing data using the *Step* scale specified above.
Display from, to	Specifies the range of data to print. If you leave the *Display from* field blank, the program uses the beginning of the data as the *Display from* value. If you leave the *to* field blank, the program uses the end of the data as the *to* value. **Note:** You can set the print range only after printing data.
Format	Specifies one of the following formats for the printed data: ■ *Engineering Suffix* (default) If you select *Engineering Suffix*, the program represents 0.0001 as 0.1m. ■ *Engineering* If you select *Engineering*, the program represents 0.0001 as 0.1e-3. ■ *Scientific* If you select *Scientific*, the program represents 0.0001 as 1e-4.
Column Width	Specifies the number of characters allowed for column width. You can specify a number from 4 to 20. The default width is 14 characters.
Column Spacing	Specifies the number of blank spaces the program uses to separate columns. You can specify a number from 1 to 10. The default spacing is 4.
Number of Significant Digits	Specifies the number of significant digits the program uses when printing results data. You can specify a number from 2 to 10. The default is 4 digits.

Displaying Untruncated Output Information

When output names are too long to fit into columns in the Results Display Window, the program truncates them. To see untruncated output names, do the following:

➤ In the Results Display Window, choose *Info – Show Output*.

The Show Output window appears.



If you have more than one output in your Results Display Window, only the last one appears in the Show Output window.

Exporting Results to a HTML or CSV File

To export results to a HTML or comma-separated values (CSV) file, do the following:

Tip

To customize the results that are exported to the HTML or CSV file, you can do the following:

- ❑ Hide the data for which you do not want to export results. For more information, see Hiding and Showing Data on the Results Tab on page 564.
- ❑ Hide columns in the *Detail - Transpose* view. For more information, see Hiding and Showing Columns in the Detail - Transpose View on page 580.
- ❑ Hide the results for the tests for which you do not want to export results. For more information, see Hiding and Showing Results for Tests on page 570.
- ❑ Only the data displayed in the Results tab is exported to the HTML or CSV file. By default, the Results tab displays the best 10 design points from the simulation run. To display more design points in the Results tab, use the `numberOfBestPointsToView` environment variable. For more information, see numberOfBestPointsToView on page 920.
- ❑ By default, results are exported to CSV files in the scientific notation format. Set the exportPreserveScalingFactors environment variable to export results in the same format as they are displayed in the Results tab to the CSV file.

1. On the Results tab of the Outputs pane, click the  button.

   The Export Results form appears.

2. In the *File name* field, type a file name with the `.html`, `.htm` or `.csv` extension.
3. Click *Save*.

   The program exports the results to the file you specified. By default, the file is saved in the `documents` folder of the ADE XL view and displayed in the *Documents* tree on the Data View pane. For more information about working with documents, see Chapter 18, “Working with Documents.”

Using SKILL to Display Tabular Data

You can use the SKILL language for queries to request other kinds of simulation results, to build output format macros, and to automate test and result reporting sequences. The syntax for queries is shown at the beginning of the line in the Results Display window.

To display…	Type this command in the CIW
A list of operating-point parameter names and their values for `R1`	`OP("/R1","??")`
A list of just the operating-point parameter names for `R1`	`OP("/R1","?")`
A single operating-point parameter (`v` for voltage, for example) and its value for `R1`	`OP("/R1","v")`
A list of transient operating-point parameter names and their values for `C1`	`OPT("/C1","??")`
A list of just the transient operating-point parameter names for `C1`	`OPT("/C1","?")`
A single transient operating-point parameter (`i` for current, for example) and its value for `C1`	`OPT("/C1","i")`
A list of model parameter names and their values for `Q1`	`MP("/Q1","??")`
A list of just the model parameter names for `Q1`	`MP("/Q1","?")`
A single model parameter (`is` for saturation current, for example) and its value for `Q1`	`MP("/Q1","is")`
Noise parameter information for a device with only one noise parameter (a resistor `R4`, for example)	`VNP("/R4")`
A list of noise parameter names for a device with more than one noise parameter (a device `D24`, for example) and their values	`VNPP("/D24","??")`
A list of just the noise parameter names for a device with more than one noise parameter (a device `D24`, for example)	`VNPP("/D24","?")`
A single noise parameter (`rs` for saturation resistance, for example) and its value for a device with more than one noise parameter (a device `D24`, for example)	`VNPP("/D24","rs")`

Annotating Simulation Results

You can annotate the schematic to show parameters, operating points, net names, and voltages of individual design components from the results displayed on the Results tab of the Outputs pane.

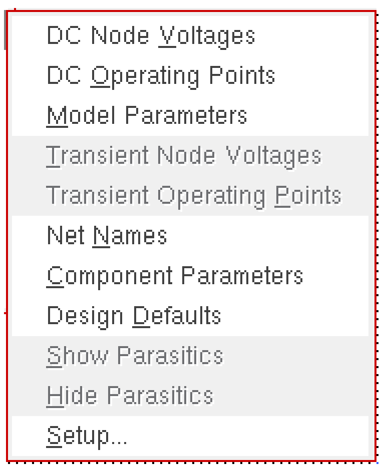
Results can be displayed by performing any one of the following actions:

- Run a simulation.
- View results.

To annotate results to the schematic, do the following:

1. On the Results tab of the Outputs pane, right-click a data point and choose *Annotate*.

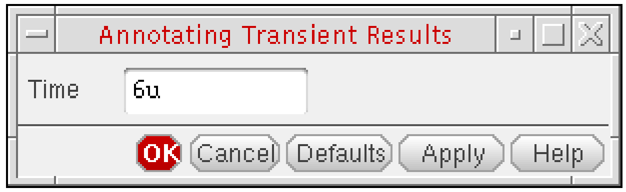
   A submenu of items that you can annotate appears. Only those items for which you have annotation data are active on the submenu.

2. Select the required item from the Annotate submenu. The following table describes each submenu item:

Annotate Submenu Item	Description
DC Node Voltages	The program annotates your design schematic with DC node voltage values.
DC Operating Points	The program annotates your design schematic with DC operating point data.

Annotate Submenu Item	Description
Model Parameters	The program annotates your design schematic with enabled model parameter values.
Transient Node Voltages	The Annotating Transient Results form appears.



1. In the *Time* field, type the transient time point.
2. Click *OK*.

   The program annotates your design schematic with transient node voltage values from the specified simulation time point.

Annotate Submenu Item	Description
Transient Operating Points	The program annotates your design schematic with operating point data for the given time point.
Net Names	The program annotates your design schematic with net names.
Component Parameters	The program annotates your design schematic with component parameters.
Design Defaults	The program restores the annotations on your design schematic from the CDF.
Show Parasitics *Hide Parasitics*	The program shows or hides parasitics when you descend (*Edit – Hierarchy – Descend Edit*) into the schematic view of a cell you simulated using its extracted view—`(out of context)` appears in the title bar **Note:** These submenu items are available only when you have DC operating point results. These submenu items also appear on the *Parasitics* menu on the main menu bar.

Annotate Submenu Item	Description
Setup	The Annotation Setup form appears. You can configure the annotation settings in the design. For more information, see Using Annotation Setup Form section in the *Virtuoso Schematic Editor L User Guide.*

To save the annotation settings, choose *File – Save* from the Annotation Setup form. For more information on saving the annotation settings, see Saving Annotation Settings section in the *Virtuoso Schematic Editor L User Guide*.

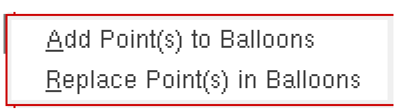
Using Annotation Balloons for Annotation

After running a simulation or viewing the results, you can annotate the simulation results from different data points using the annotation balloons on the schematic canvas. You can set multiple data points for annotations from different data points from a history and from different histories. The annotated data is displayed in the tabular form on the annotation balloons. As a result, you can compare and debug data from multiple data points.

To annotate data points on the annotation balloons, do the following:

1. On the Results tab of the Outputs pane, right-click a data point and choose *Annotation Balloons*.

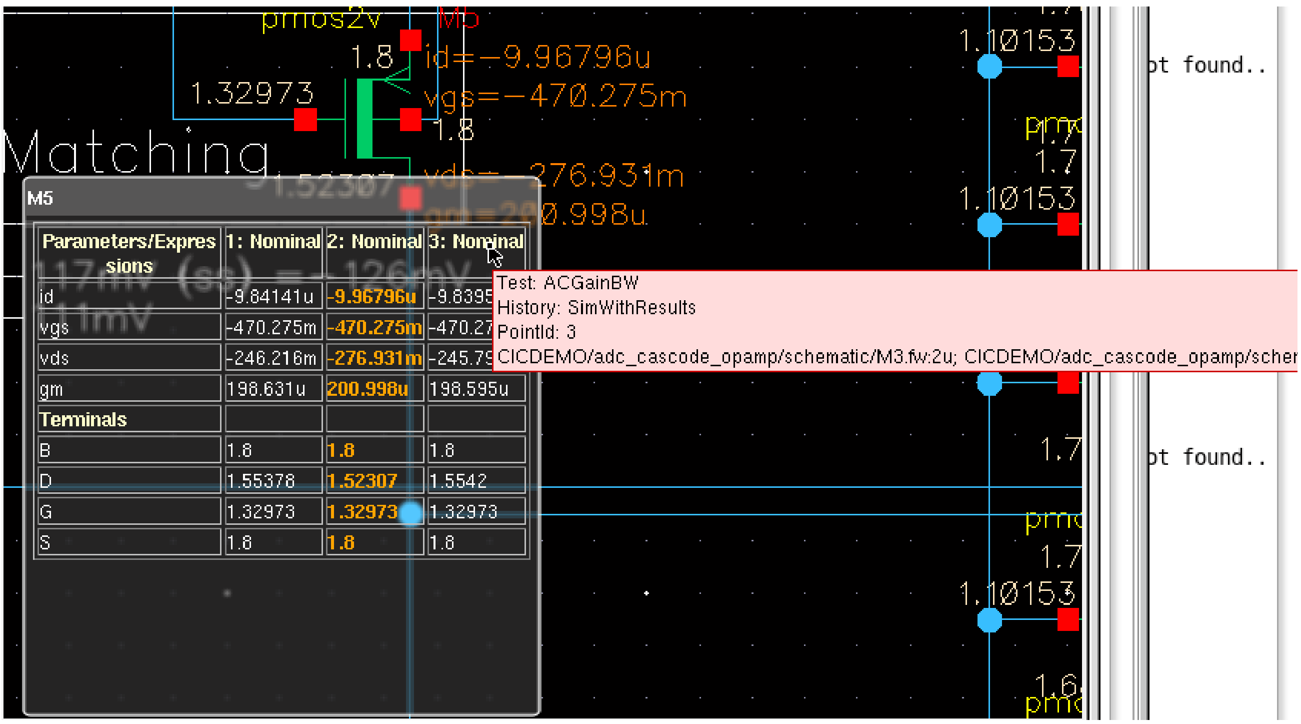
   A submenu of options appears.



2. Select an available option from the *Annotation Balloons* submenu:

Annotate Submenu Item	Description
Add Point(s) to Balloons	Adds the values of the selected data point to the existing data on the annotation balloon and displays the data in a tabular form.
Replace Point(s) in Balloons	Replaces the value of the last selected data point with the existing data on the annotation balloon.

Annotation balloons are displayed when you move the cursor over any instance on the schematic. When you add multiple data points for annotation, the values of the last selected data point are annotated on the schematic as well as displayed on annotation balloons. As a result, the last selected data point result is highlighted in different color on annotation balloon to differentiate from the other data point results. In the annotation balloon, the color of the data point annotated on canvas appears yellow when the transparency is set from 21 to 65; and orange when the transparency is set from 0 to 20, or from 66 to 100. You can change the color settings of the annotation balloon using the Annotation Balloon Options.



The annotation balloon shows the following information in the tooltip of the column header, when displaying information for multiple data points:

- Test name
- History name
- Data point ID
- Sweep parameters

The header of the annotation balloon shows the following information:

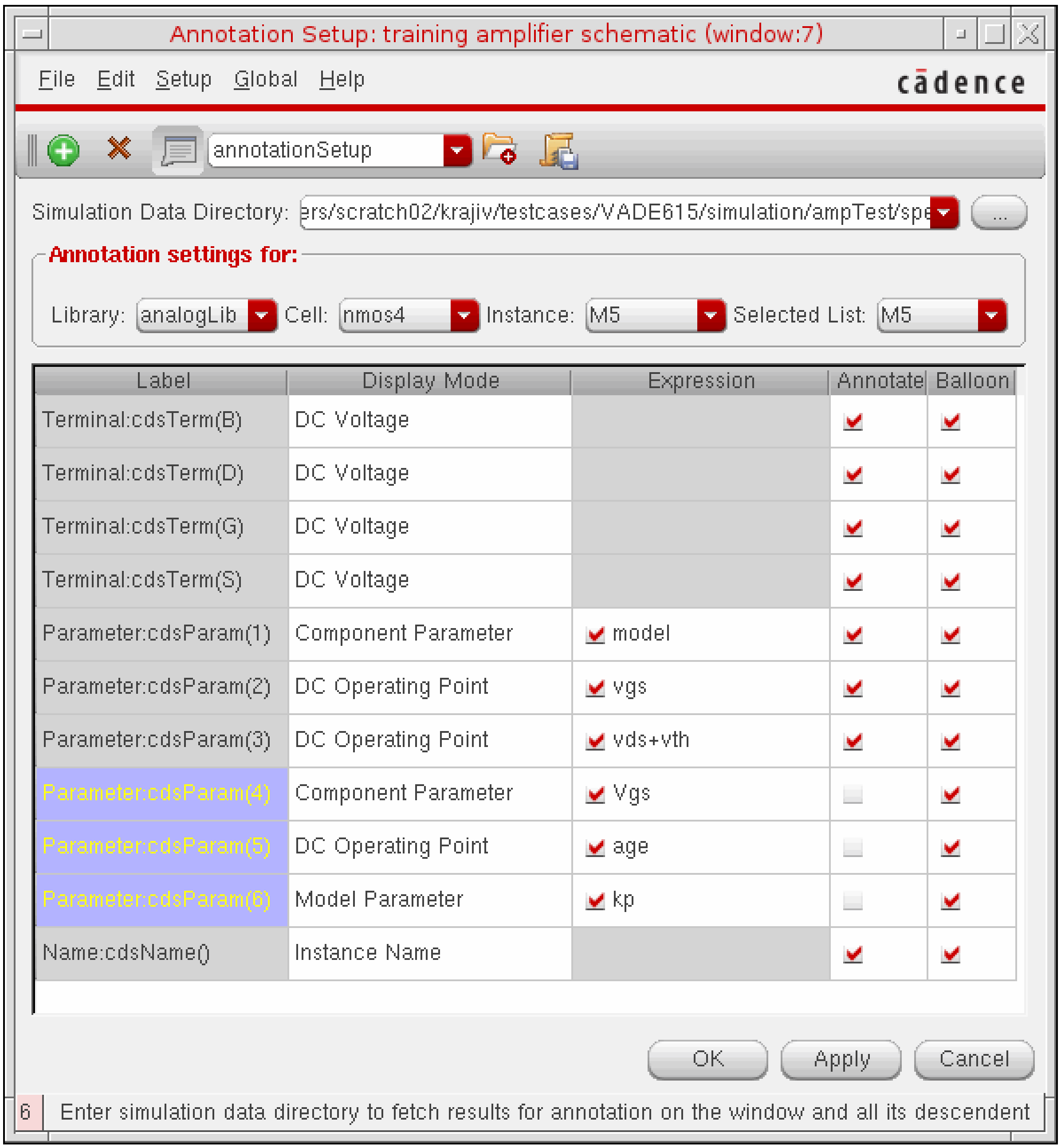
- Design point ID
- Corner name

Specifying the Data Directory for Labels

To specify the simulation data directory (run directory) for labels, do the following:

1. In the schematic window, choose *View – Annotations – Setup.*

   The Annotation Setup form appears.



2. In the *Simulation Data Directory* field, type the path to the simulation run directory.

3. Click *OK*.

Note: You do not need to use this form if

- The analog circuit design environment is active and you have specified the correct directory as the run directory
- You used the Results Browser to select results for the current schematic
- The most recent simulation you ran was for this schematic

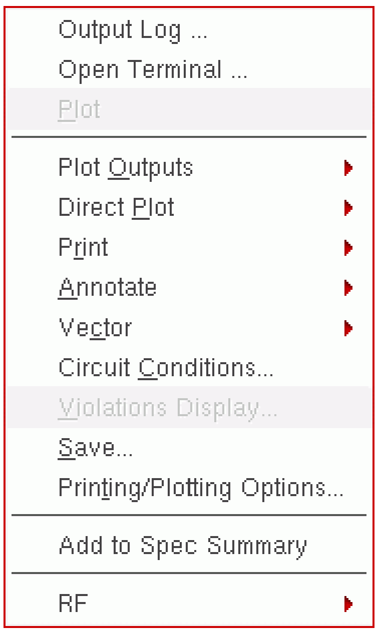
Viewing Results from the Data View Pane

You can view results from various checkpoints listed in the History tab of the Data View pane. See "Viewing Results from a Particular Checkpoint" on page 816 for more information.

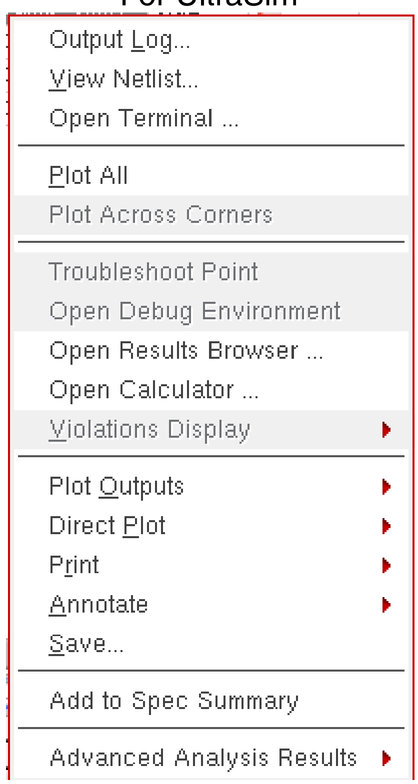
Results Tab Right-Click Menus

When you right-click a test name on the Results tab of the Outputs pane, a pop-up menu appears.
The items on the pop-up menu depend on the simulator:

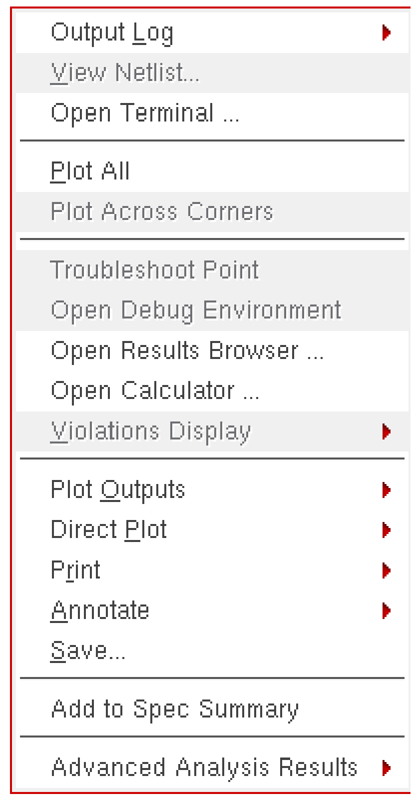
For Spectre



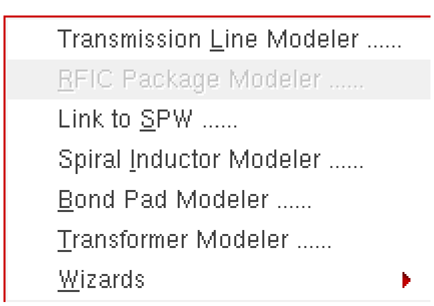
For UltraSim



For AMS

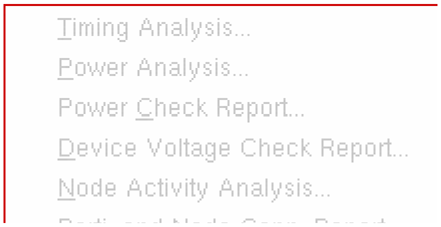


For the Spectre circuit simulator, the following items appear on the *RF* submenu:



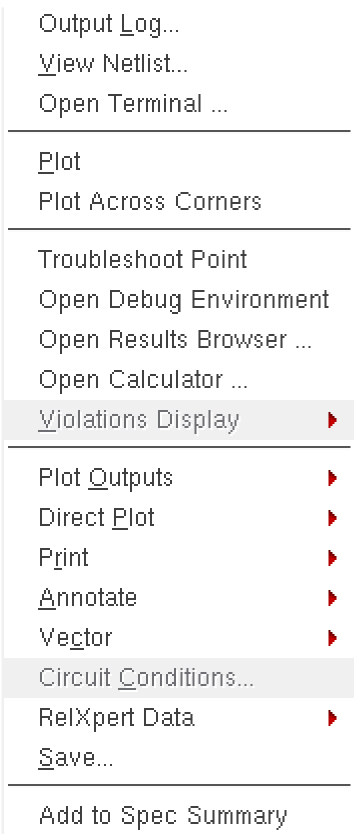
For the UltraSim and AMS circuit simulators, the following items appear on their respective *Advanced Analysis Results* submenus:

For UltraSim



For AMS

When you right-click an output on the Results tab of the Outputs pane, the following pop-up menu appears:



15

Working with Specifications

You define performance specifications on the Outputs Setup tab of the Outputs pane. After simulation, the program displays information so that you can see whether your design met, nearly met, or failed to meet your performance specifications.

You can view specification (spec) results from one or more history items (checkpoints) on the Results tab of the Outputs pane. You can select which measurement result for which corner or sweep to display. The program displays specification information such as the corner or sweep name, the analysis name, the measurement name, the conditions, the minimum or maximum values. You can view the spec for each measurement along with information about whether each spec passed or failed. You can create a summary report of pass/fail results.

You can control the format for each specification such that the program uses particular units or engineering or scientific notation when displaying each spec. You can sort by measurement name and export your spec sheet to a comma-separated values file.

See the following topics for details:

Working with Specifications

The outputs of type Expression, OCEAN Script, MATLAB Script, Area Goal, Device Check or Op Region Spec that are added in the Outputs Setup tab of the Outputs pane are called measurements.

You can define a performance specification for a measurement in the *Spec* column on the Outputs Setup tab of the Outputs pane or in the Override Specifications form.

By default, the specification defined for a measurement applies to all the corners enabled for the test. You can override the measurement specification for a corner or disable the corner specification.

For more information, see the following topics:

Defining a Specification

To define a specification for a measurement on the Outputs Setup tab, do the following:

1. On the Outputs Setup tab of the Outputs pane, double-click in the *Spec* column.

   The *Spec* cell has two parts: a specification type (which you select from a drop-down list) and a target value.

2. In the drop-down list that appears in the left half of the cell, select one of the following specification types and fill in the right half of the cell with appropriate target information as indicated:

Table 15-1 Specification Types

Objective	Target	Description
minimize	*targetValue*	Minimize the result; optional target value
maximize	*targetValue*	Maximize the result; optional target value
<	*targetValue*	The result must be less than the target value
>	*targetValue*	The result must be greater than the target value
range	*lowerBound upperBound*	The result must fall between the boundary values
tol	*targetValue percentageDeviation*	The result must be within the specified percentage deviation of the target value
info		Choose this setting when you want the measured value to appear but there is no target value or performance objective to meet

3. (Optional) In the Weight column next to the specification, specify the weighting factor for the specification.

To define a specification for a measurement in the Override Specifications form, do the following:

1. Right-click on the row for a measurement in the Outputs Setup tab of the Outputs pane and choose *Override Specifications* from the pop-up menu.

The Override Specifications form appears.

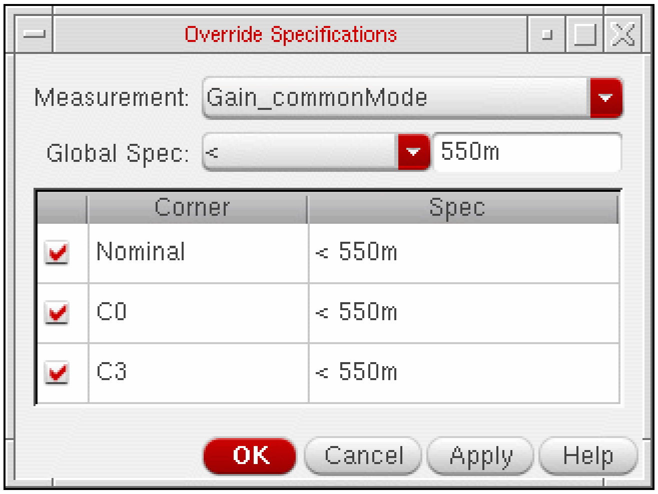


Table 15-2 Override Specifications Form Field Description

Field	Description
Measurement	Displays the measurements for which you can define specifications. Note the following: ■ By default, the name of the measurement you selected on the Outputs Setup tab of the Outputs pane is displayed. ■ This drop-down list displays only the names of the measurements for the test whose measurement you selected on the Outputs Setup tab. To define a spec for a measurement of another test, right-click on a measurement for that test and choose *Override Specifications*. ■ If you have not specified the name for a measurement in the *Name* column on the Outputs Setup tab, the expression for that measurement is displayed as its name in this drop-down list.
Global Spec	Displays the specification for the measurement. To define or modify a specification, select one of the specification types (described in Table 15-1 on page 703) from the drop-down list, and fill in the text field next to the drop-down list with appropriate target information for the measurement.

Table 15-2 Override Specifications Form Field Description

Field	Description
Corner	Displays the list of corners enabled for the test. For more information about enabling corners for tests, see Disabling and Enabling Corners on page 242. You can disable or enable a corner specification for a specific measurement. For more information, see Disabling and Enabling Corner Specifications on page 710
Spec	Displays the specification for each corner enabled for the test. For more information about specifying a different specification for a corner, see Overriding the Measurement Specification for a Corner on page 707.

2. In the *Measurement* drop-down list, select the measurement for which you want to define the specification.

3. In the *Global Spec* drop-down list, select one of the specification types (described in Table 15-1 on page 703) from the drop-down list, and fill in the text field next to the drop-down list with appropriate target information for the measurement.

4. Click *OK*.

   By default, the specification applies to all the corners selected for the test. You can do the following:

   - Override the specification for a specific corner. For more information, see Overriding the Measurement Specification for a Corner on page 707.
   - Undo a corner specification override. For more information, see Undoing a Corner Specification Override on page 710.
   - Disable or enable the corner specification for a specific measurement. For more information, see Disabling and Enabling Corner Specifications on page 710.
   - Disable or enable the corner specification across all measurements for a test. For more information, see Disabling and Enabling Corner Specifications on page 710.

The next time you run a simulation, the program measures the result against the performance specification and displays the results in the Results tab of the Outputs pane. For more information about how results are displayed in the Results tab, see Viewing Specification Results in the Results Tab on page 737.

Tip

If you add or modify measurement expressions or specifications after running a simulation, you can click the button on the Results tab to update the results displayed in the Results tab based on the new or modified expressions and specifications. For more information about using the button, see Re-evaluating Expressions and Specifications on page 650.

The results will also be updated based on the new or modified expressions and specifications if you click the button on the Results tab to plot across all points. For more information about using the button, see Plotting Results on page 624.

Note: For information about setting up specifications for optimization, see "Setting Up Specifications" in the *Virtuoso Analog Design Environment GXL User Guide*.

See also:

- Starting a Simulation on page 462
- Re-evaluating Expressions and Specifications on page 650
- Viewing Specification Results in the Results Tab on page 737

Copying a Specification

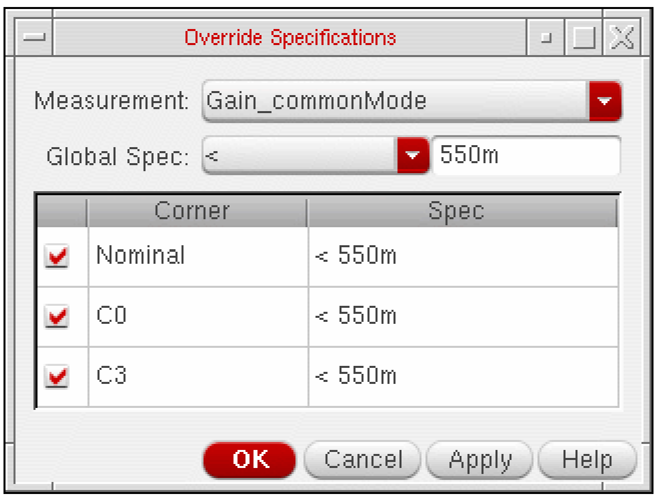
To copy a specification, do the following:

1. In the *Spec* column on the Outputs Setup tab of the Outputs pane, select the specification.
2. Press *Ctrl+C* to copy the specification.
3. Click in the *Spec* column next to the expression for which you want to paste the specification.
4. Press *Ctrl+V* to paste the specification.

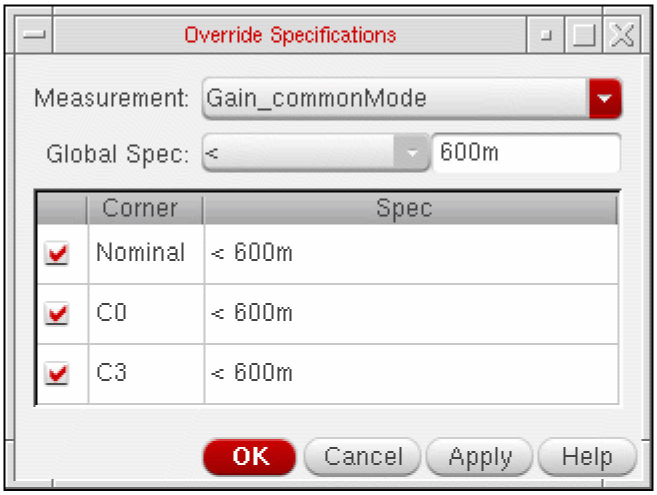
Overriding the Measurement Specification for a Corner

By default, the specification defined for a measurement applies to all the corners enabled for the test. For example, assume that the nominal corner and two corners named `C0` and `C3` are enabled for a test named `myTest`. If you define a measurement named `Gain_commonMode` for the test, the specification for the `Gain_commonMode` measurement applies to the nominal corner and to corners `C0` and `C3`.

For example, in the following figure, the specification `<550m` for the `Gain_commonMode` measurement applies to all the corners displayed in the *Corner* column.



If you modify the specification for a measurement in the Outputs Setup tab of the Outputs pane or in the *Global Spec* field on the Override Specifications form, the change will apply to all the corners selected for the test. For example, the following figure indicates that when the specification for the `Gain_commonMode` measurement is changed from `<550m` to `<600m`, the change is applied to all the corners displayed in the *Corner* column.



You can override the measurement specification for a corner by specifying a different specification for the corner.

To override the measurement specification for a specific corner, do the following:

1. Right-click on the row for a measurement in the Outputs Setup tab of the Outputs pane and choose *Override Specifications* from the pop-up menu.

   The Override Specifications form appears.

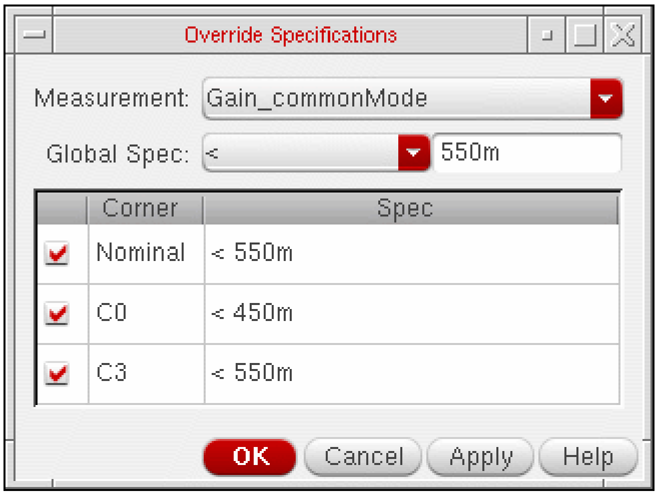
2. In the *Measurement* drop-down list, select the measurement for which you want to override a corner specification.

3. In the *Spec* column, double-click on the specification for the corner.

   The *Spec* cell has two parts: a specification type (which you select from a drop-down list) and a target value.

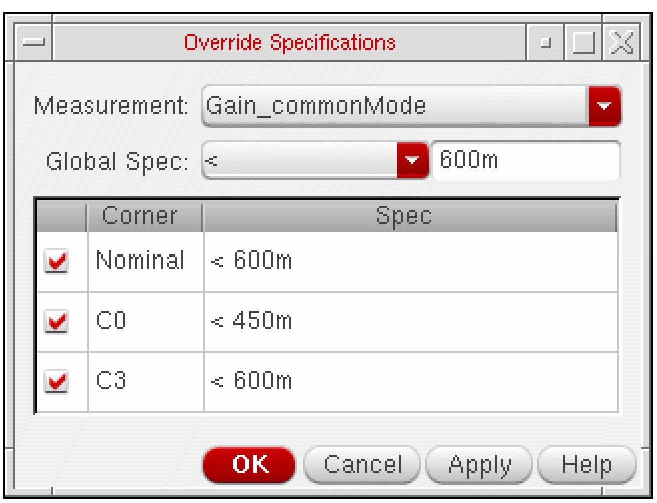
4. In the drop-down list that appears in the left half of the cell, select one of the specification types described in Table 15-1 on page 703 and fill in the right half of the cell with appropriate target information for the corner.

For example, in the following figure, the specification `<450m` for corner `C0` overrides the specification `<550m` for the `Gain_commonMode` measurement.



5. Click *OK*.

Note: If you later modify the specification for a measurement in the Outputs Setup tab of the Outputs pane or in the *Global Spec* field on the Edit Specification form, the change will apply only to the corners for which you did not override the specification. For example, the following figure indicates that when the specification for the `Gain_commonMode` measurement is changed from `<550m` to `<600m`, the change is applied only to the `Nominal` corner and corner `C3`. The change is not applied to corner `C0` because the specification `<450m` for corner `C0` overrides the specification for the `Gain_commonMode` measurement.



See also:

Disabling and Enabling Corners on page 242

Viewing Specification Results in the Results Tab on page 737

Undoing a Corner Specification Override

To undo a corner specification override, do the following in the Override Specifications form:

1. In the *Measurement* drop-down list, select the measurement for which you want to undo a corner specification override.
2. In the *Spec* column, right-click on the specification for the corner for which you want to undo the specification override and choose *Set to Global* from the pop-up menu.

   To undo the overrides for multiple corners, hold down the *Shift* key (for contiguous selection) or the *Ctrl* key (for noncontiguous selection), right-click and then choose *Set to Global* from the pop-up menu.

   The specification for the measurement (displayed in the *Global Spec* field) is applied to the corner.

Disabling and Enabling Corner Specifications

By default, the specification defined for a measurement applies to all corners enabled for the test. You can do the following:

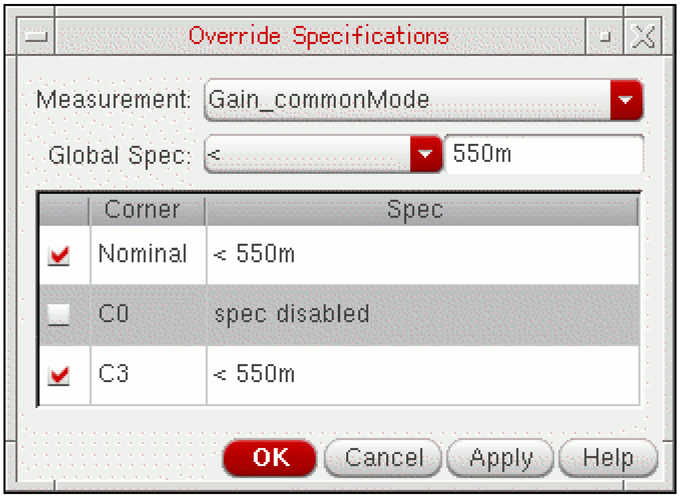
- Disable or enable the corner specification for a specific measurement.
- Disable or enable the corner specification across all measurements for a test.

To disable or enable a corner specification for a specific measurement, do the following:

1. Right-click on the row for a measurement in the Outputs Setup tab of the Outputs pane and choose *Override Specifications* from the pop-up menu.

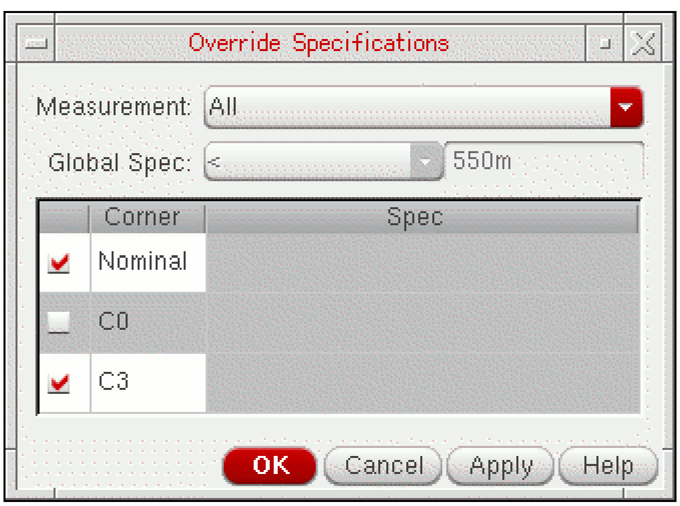
   The Override Specifications form appears.
2. In the *Measurement* drop-down list, select the measurement for which you want to disable or enable the corner specification.
3. Do one of the following:
   - To disable a corner specification for the measurement, clear the check box next to the corner.

For example, in the following figure, the specification for corner `C0` is disabled.



- ❑ To enable a corner specification for the measurement, select the check box next to the corner.
- ❑ To disable a corner specification for all the measurements for the test, select *All* in the *Measurement* drop-down list, then clear the check box next to the corner.

  For example, in the following figure, the specification for corner `C0` is disabled for all the measurements for the test.



- ❑ To enable a corner specification for all the measurements for the test, select *All* in the *Measurement* drop-down list, then select the check box next to the corner.

Note: You can also right-click on the first column in the Override Specifications form and choose *Disable All* or *Enable All* to disable or enable all the corner specifications.

4. Click *OK*.

See also:

Viewing Specification Results in the Results Tab on page 737Defining Operating Region Specifications

Operating region specifications define expressions that ensure that your devices are operating in a desired region (saturation, triode, and so on).

For more information, see the following topics:

- Setting Up Operating Region Specifications on page 712
- Saving Operating Region Expressions to a File on page 733
- Loading Operating Region Expressions from a File on page 735
- Modifying Operating Region Specifications on page 729
- Deleting Operating Region Specifications on page 736
- Disabling and Enabling Operating Region Specifications on page 736
- Viewing Operating Region Violations on page 741
- Re-evaluating Operating Region Expressions on page 744

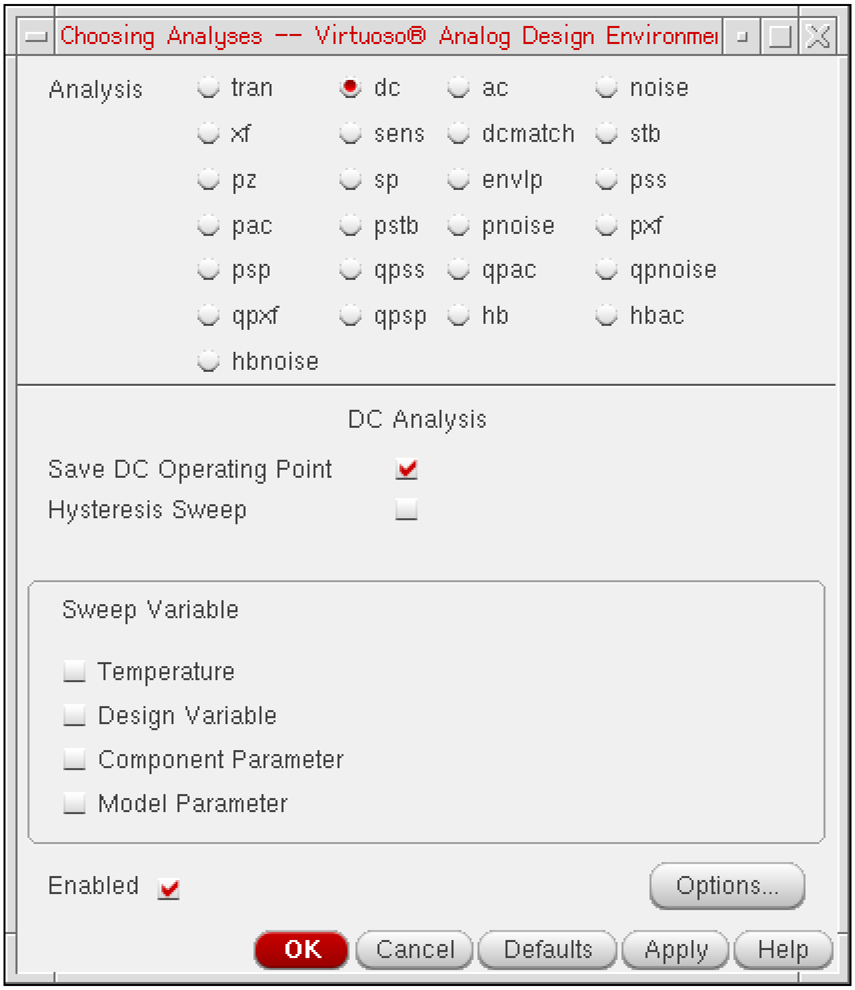
Important

In the IC6.1.4 and previous releases of ADE XL, operating region expressions for each operating region specification were defined as dcOp device checks. However, from the IC6.1.5 release, ADE XL provides a separate user interface for defining operating region expressions. Therefore, you need to migrate the operating region expressions that were defined in IC6.1.4 to IC6.1.5. For more details, refer to Migrating Operating Region Specifications from IC6.1.4 to IC6.1.5 on page 748.

Setting Up Operating Region Specifications

To set up an operating region specification for a test, do the following:

1. Ensure that a DC analysis with option to save DC operating point information is set up for the test as shown below:

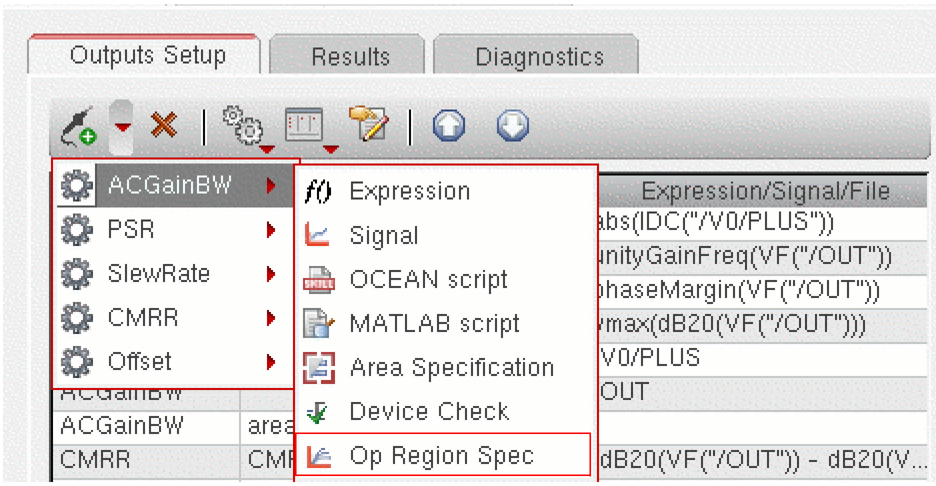


For more information about setting up a DC analysis, see Adding an Analysis.

Note: In the earlier releases, for operating region specifications, it was required to enable the *dochecklimit* device checking option in the Simulator Options form. Starting from IC6.1.5 ISR8, the operating region specifications do not depend on device checks and it is not required to check this option.

2. On the Outputs Setup tab, click the *Add new output*  toolbar button, to view the drop-down list of available tests.

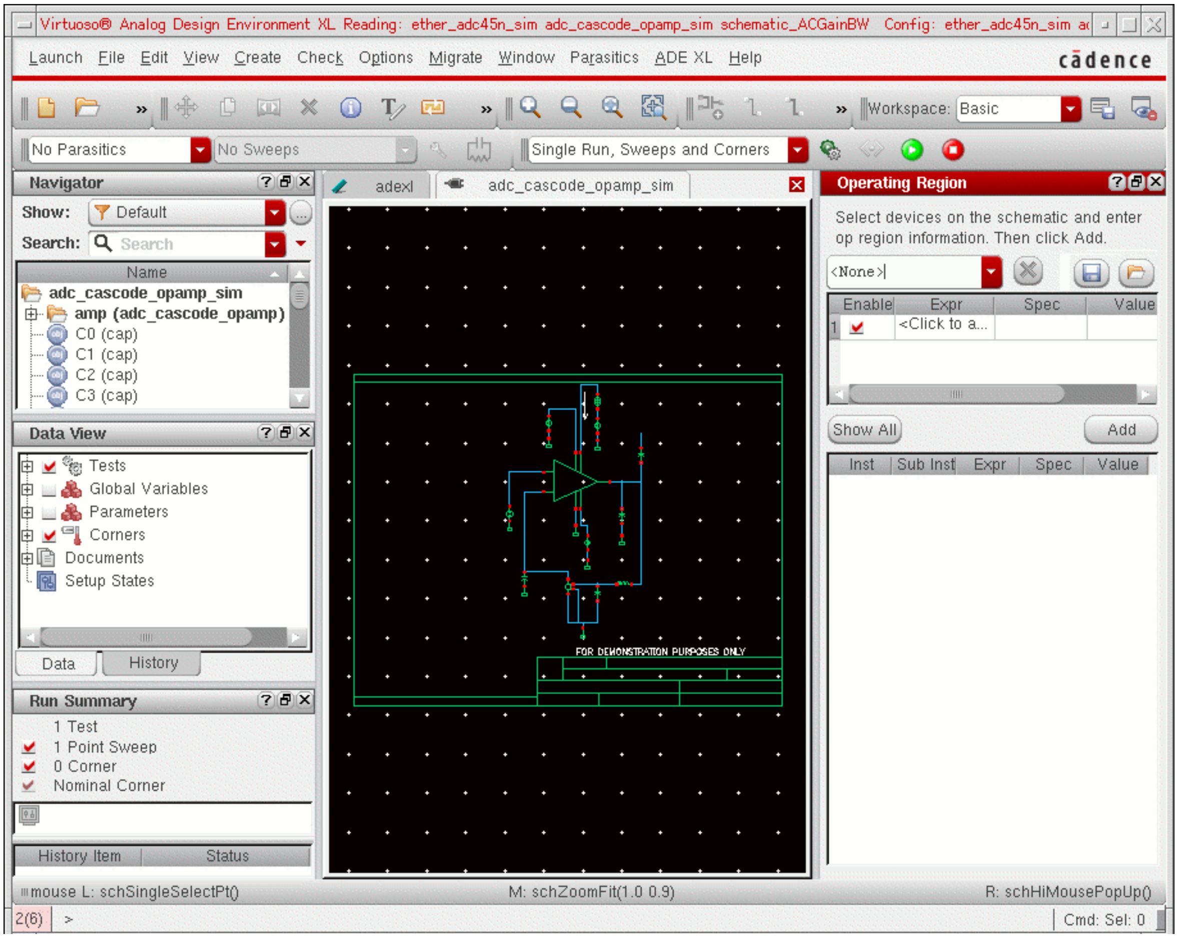
3. In the drop-down list, select a test and choose *Op Region Spec*.



Tip

You can also right-click a test name in the Outputs Setup tab and choose *Add Op Region Spec* to set up an operating region specification for that test.

The schematic for the test is displayed with the Operating Region and Navigator assistant panes docked.



4. Specify operating region expressions using the Operating Region assistant pane or the Operating Region Specification form.

   The Operating Region assistant pane enables you to quickly specify expressions while the Operating Region Specification form provides advanced methods for specifying expressions.

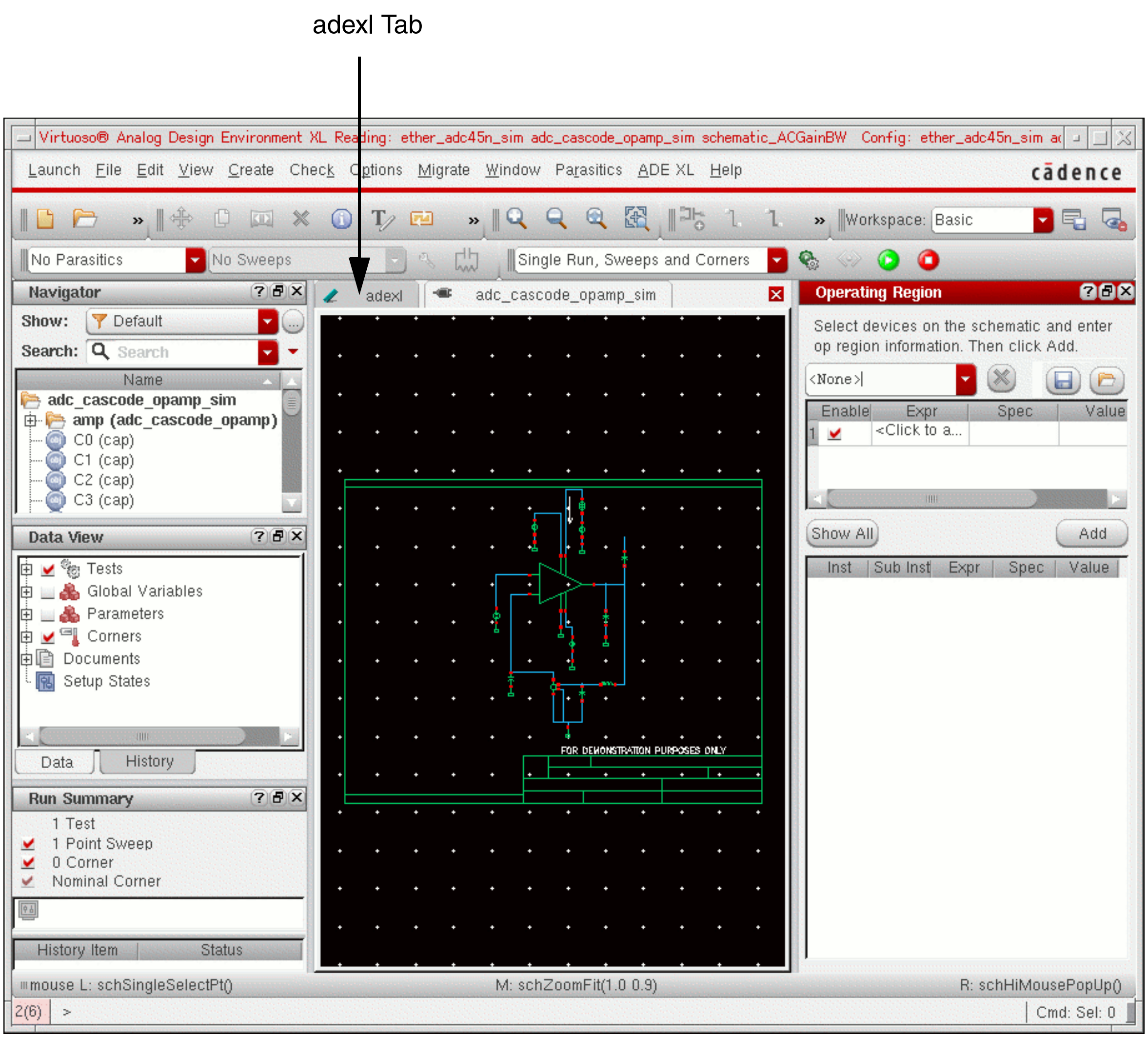
   Note: You can specify multiple operating region specifications for a single test.

   For more information, see the following topics:

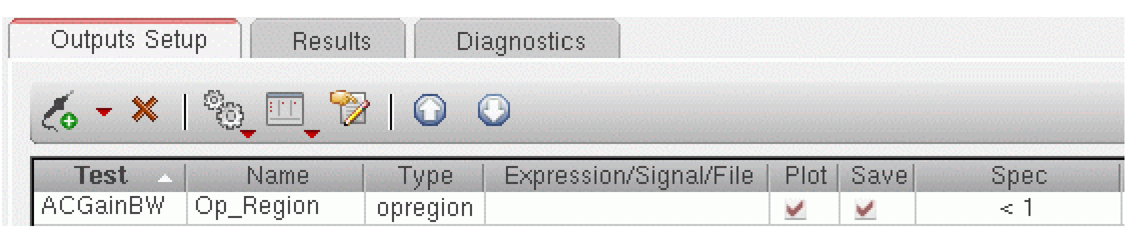
   - Specifying Operating Region Expressions Using the Operating Region Assistant Pane on page 717
   - Specifying Operating Region Expressions Using the Operating Region Specification Form on page 722

- ❑ Viewing Operating Region Violations on page 741
- ❑ Re-evaluating Operating Region Expressions on page 744

5. Click the adexl tab when you are done.



The operating region specification for the test is displayed in the Outputs Setup tab as shown below:



Specifying Operating Region Expressions Using the Operating Region Assistant Pane

To specify operating region expressions using the Operating Region assistant pane, do the following:

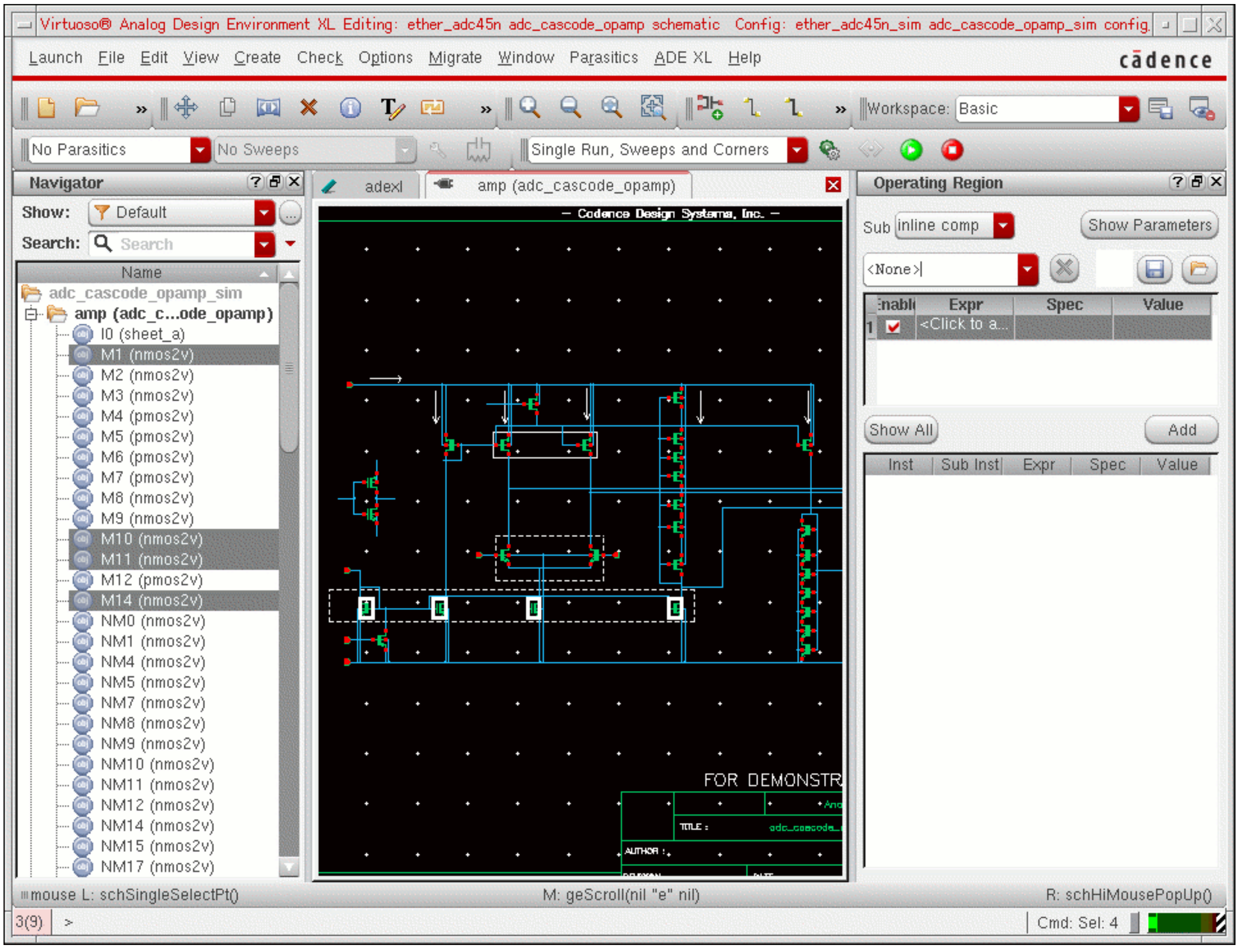
1. On the schematic or in the Navigator assistant pane, select the instance for which you want to specify operating region expressions.

   You can also select more than one instance of the same type to simultaneously specify operating region expressions for all of them.

   - To select more than one instance at a time on the schematic, do one of the following:
     - Hold down the *Shift* key and click on instances.
     - Click and drag the mouse over the instances you want to select.

       All the instances that are within the yellow bounding box that appears are included in the selection.
   - To select more than one instance at a time in the Navigator assistant pane, hold down the *Shift* key (for contiguous selection) or the *Ctrl* key (for noncontiguous selection) and click the next instance to add it to the selection set.

For example, in the following figure, four instances of the `nmos2v` device are selected in the Navigator assistant pane.



2. Specify operating region expressions for the selected instances by doing the following in the Operating Region assistant pane:

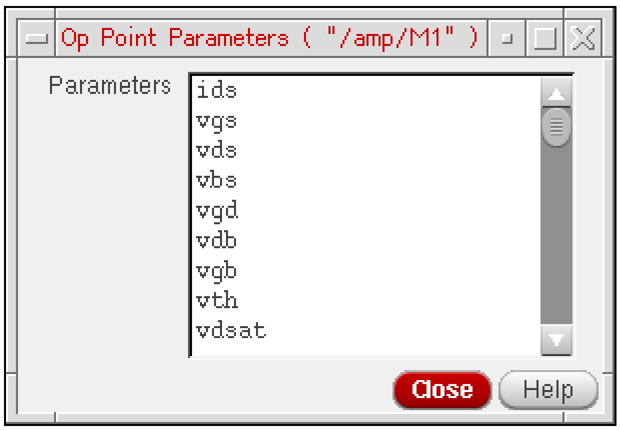
a. (Optional) If the selected instance has subcircuit instances, select the subcircuit instance for which the operating region expression is to be applied from the *SubInst* drop-down list.

b. In the *Expr* column on the Operating Region assistant, click where it says *<Click to add>* and enter an operating region expression. For example:

```
vgs - vth
```

Note: To view the list of operating point parameters that you can use in expressions, click the *Show Parameters* button on the Operating Region assistant.

Note: The operating point parameters for the selected instances are displayed in the Op Point Parameters form.

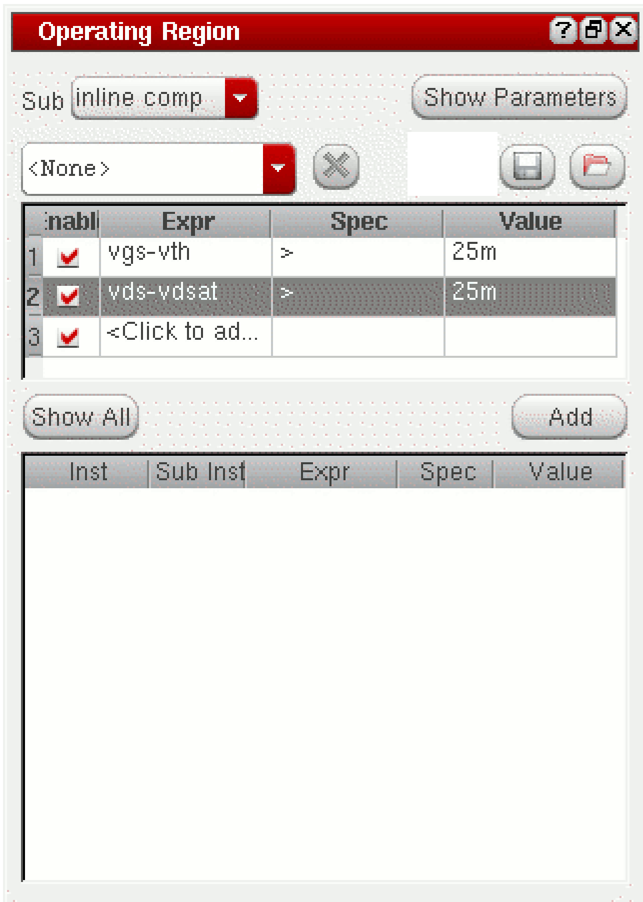


Identify the parameters from this list and create expression in the *Expr* column of the Operating Region assistant.

c. Click in the *Spec* column and select the specification for the expression.

d. Click in the *Value* column and specify the target value for the specification.

For example, in the following figure, two expressions have been specified for the selected instances.



3. By default, the specified expressions are enabled. Ensure that the expressions you want to add for the selected instances are enabled, and the expressions you do not want to add for the selected instances are disabled.

- ❑ To enable an expression, select the check box next to it in the *Enable* column.
- ❑ To disable an expression, deselect the check box next to it in the *Enable* column.

4. Click *Add*.

The enabled expressions are added for the selected instances and displayed in the table at the bottom of the Operating Region assistant pane as shown below:

Note: The Operating Region assistant form displays only the operating region expressions specified for the instances that are currently selected on the schematic or in the Navigator assistant pane. However, if no instance is selected, expressions for all the instances are displayed in the Operating Region assistant. Alternatively, you can click the *Show All* button to view all the operating region expressions specified for the test in the Operating Region Specification form. For more information about using the Operating Region Specification form, see Modifying Operating Region Specifications on page 729.

See also:

- Saving Operating Region Expressions to a File on page 733
- Loading Operating Region Expressions from a File on page 735
- Copying and Pasting Operating Region Expressions on page 732
- Deleting Operating Region Expressions on page 732

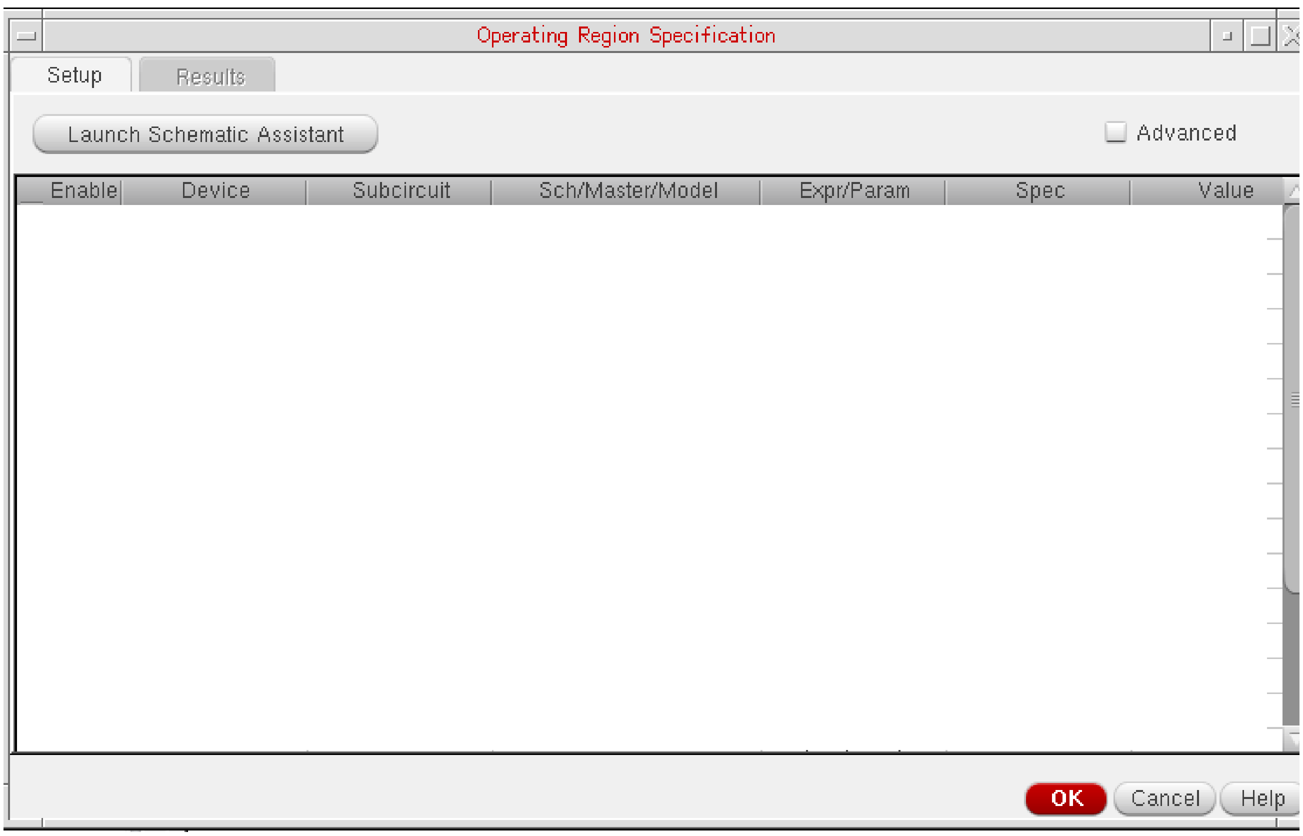
Specifying Operating Region Expressions Using the Operating Region Specification Form

You can use the advanced view of the Operating Region Specification form to specify operating region expressions.

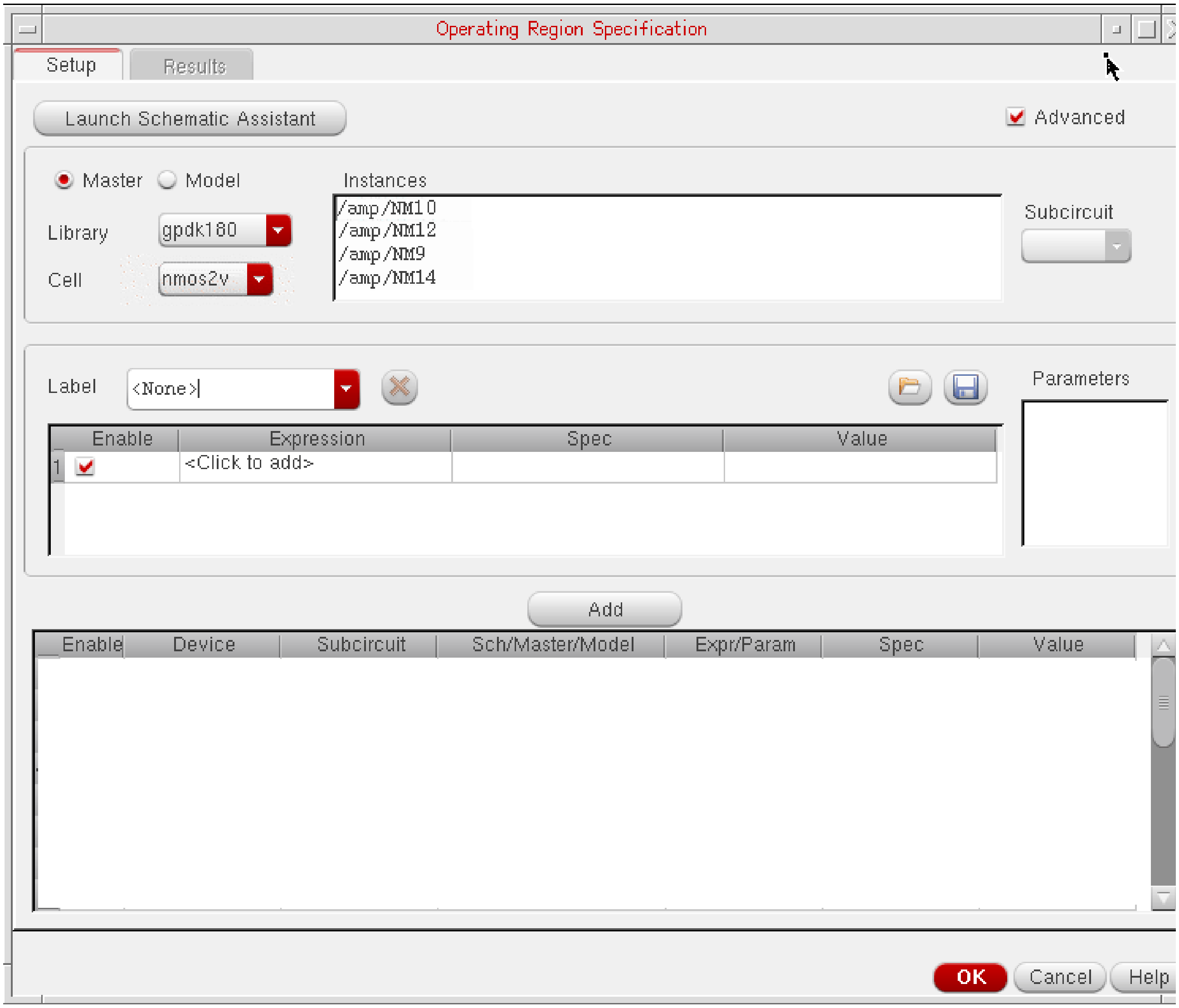
To specify operating region expressions using the Operating Region Specification form, do the following:

1. Click the *Show All* button on the Operating Region assistant pane.

   The Operating Region Specification form appears.



2. Click the *Advanced* check box to display the advanced view of the form.

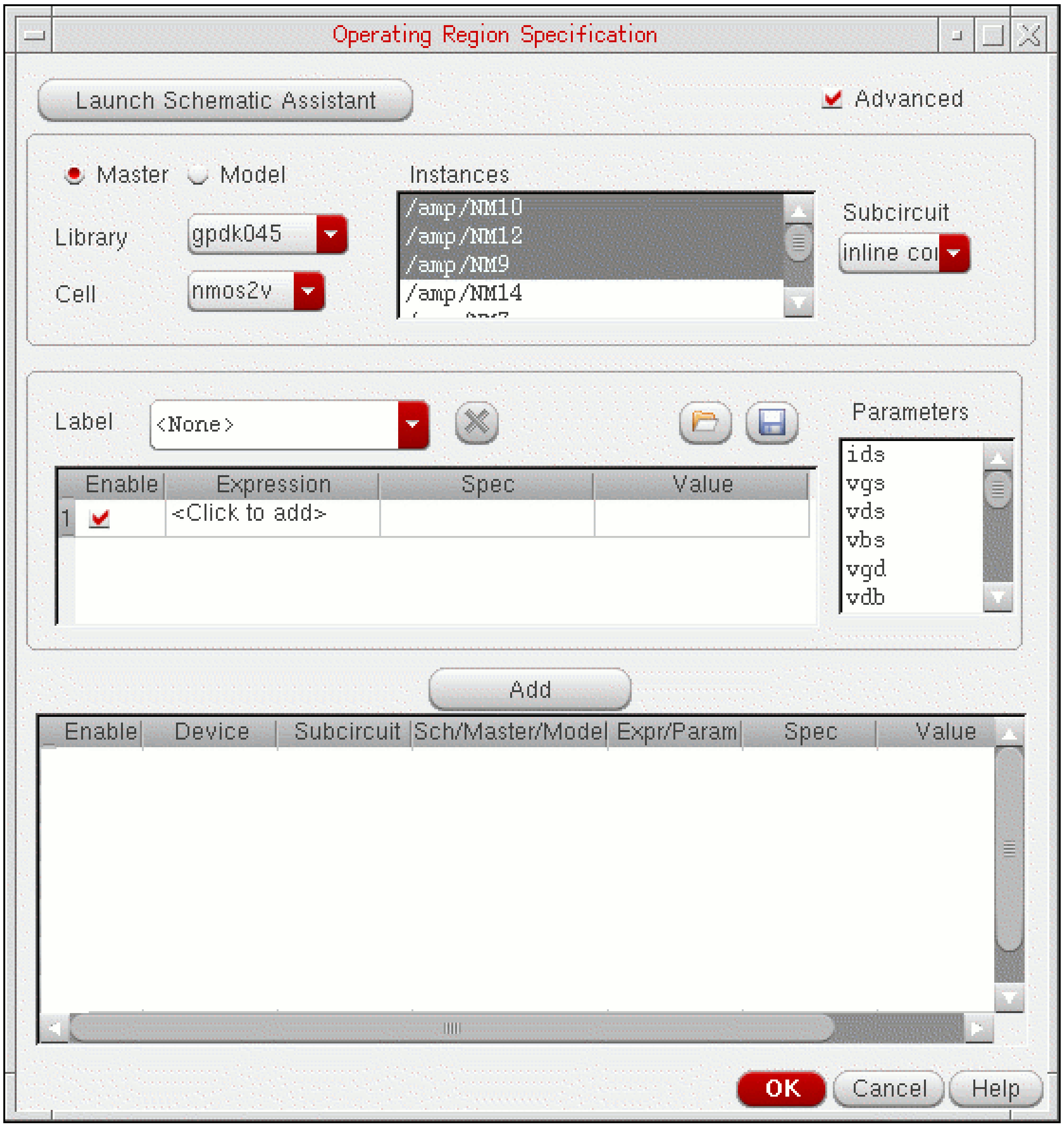


3. Select the instances for which you want to add operating region expressions by doing one of the following:

Select	**To**
Master	Select instances based on the library in which they exist. Then, do the following to select instances: **1.** Use the *Library* and *Cell* drop-down lists to select the library and cell in which cellviews for instances in the design for the test exist. All the instances of the cellviews in the design for the test are displayed in the *Instances* list box. **2.** In the *Instances* list box, select the instances for which you want to add operating region expressions. To select more than one instance at a time, hold down the *Shift* key (for contiguous selection) or the *Ctrl* key (for noncontiguous selection) and click the next instance to add it to the selection set. **3.** (Optional) If the selected instance has subcircuit instances, select the subcircuit instance for which the operating region expression is to be applied from the *Subcircuit* drop-down list.
Model	Select instances based on the models specified for the instances. Then, do the following to select instances: **1.** From the *Model* drop-down list, select a model. All the instances in the design for the test for which the model is specified are displayed in the *Instances* list box. **2.** In the *Instances* list box, do one of the following: ❑ Select *All* to add operating region expressions to all instances for which the model is specified. ❑ Select the specific instances for which you want to add operating region expressions. To select more than one instance at a time, hold down the *Shift* key (for contiguous selection) or the *Ctrl* key (for noncontiguous selection) and click the next instance to add it to the selection set.

Tip

The list of operating point parameters that you can use in expressions for the selected instances are displayed in the *Parameters* field.



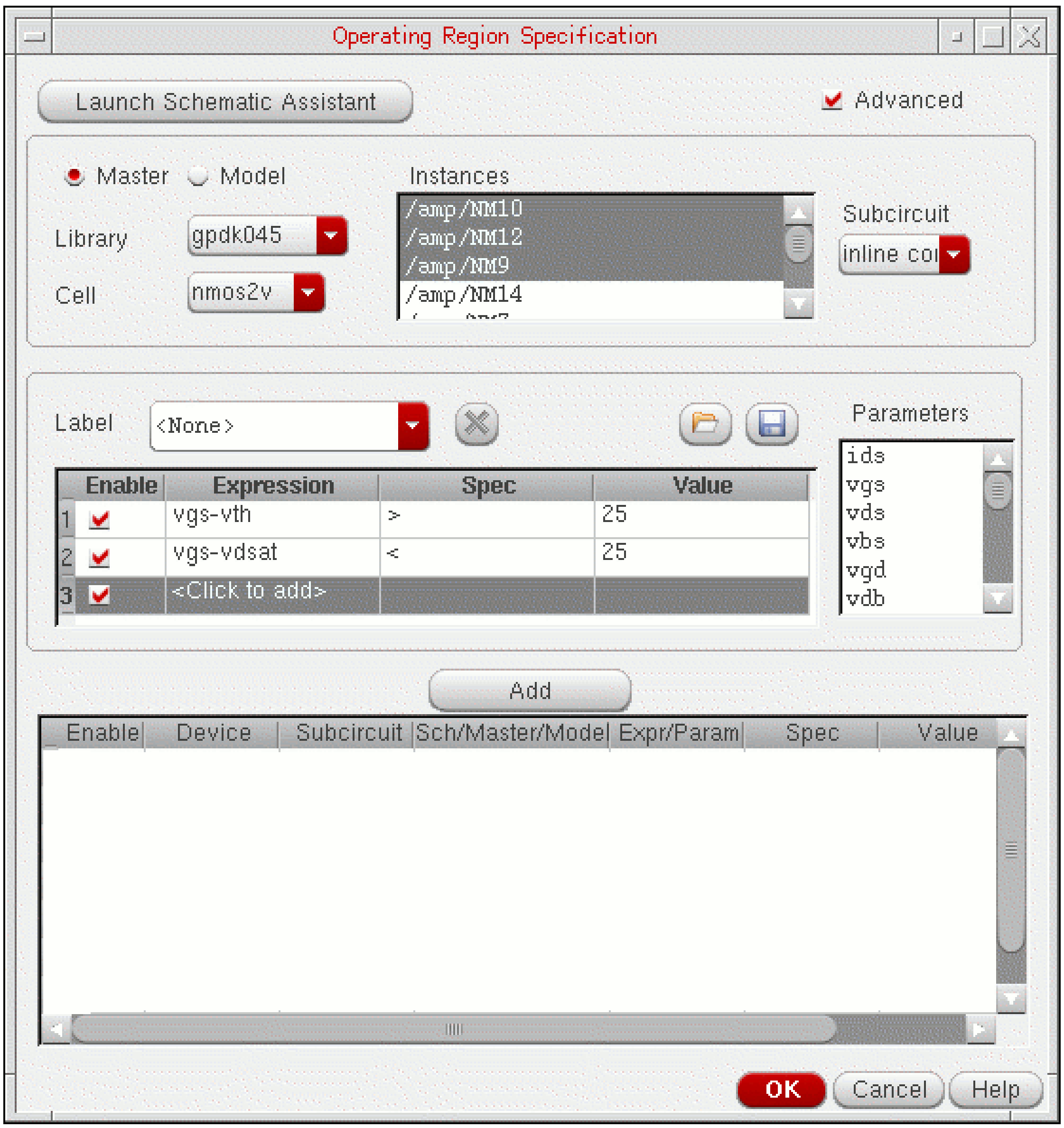
4. Specify operating region expressions for the selected instances by doing the following:

a. In the *Expression* column, click where it says *<Click to add>* and enter an operating region expression.

b. Click in the *Spec* column and select the specification for the expression.

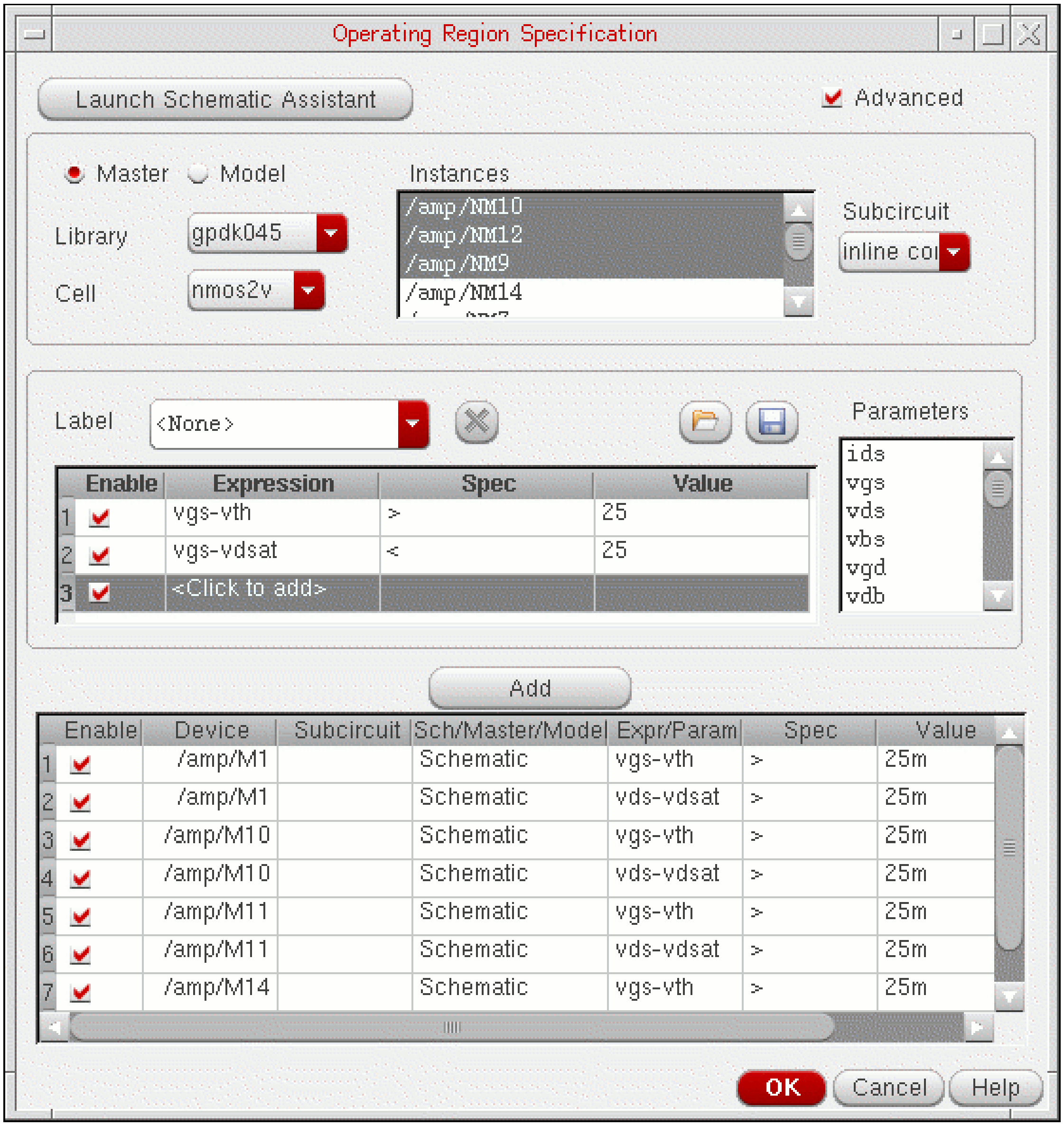
c. Click in the *Value* column and specify the target value for the specification.

For example, in the following figure, two expressions have been specified for the selected instances.



5. By default, the specified expressions are enabled. Ensure that the expressions you want to add for the selected instances are enabled, and the expressions you don't want to add for the selected instances are disabled.
   - To enable an expression, select the check box next to it in the *Enable* column.
   - To disable an expression, deselect the check box next to it in the *Enable* column.
6. Click *Add*.

   The enabled expressions are added for the selected instances and displayed in the table at the bottom of the Operating Region Specification form as shown below.



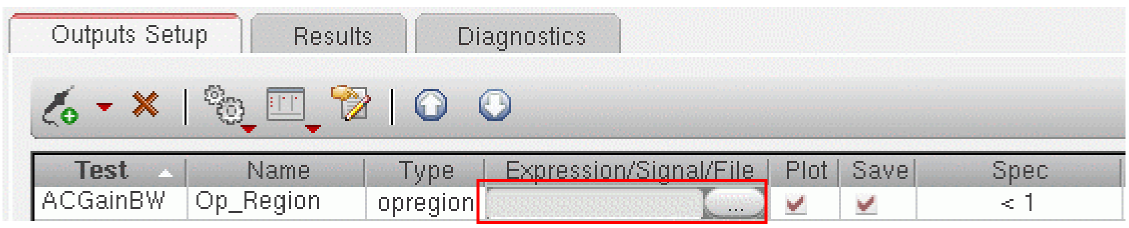
See also:

- Saving Operating Region Expressions to a File on page 733
- Loading Operating Region Expressions from a File on page 735
- Disabling and Enabling Operating Region Expressions on page 732
- Copying and Pasting Operating Region Expressions on page 732
- Deleting Operating Region Expressions on page 732

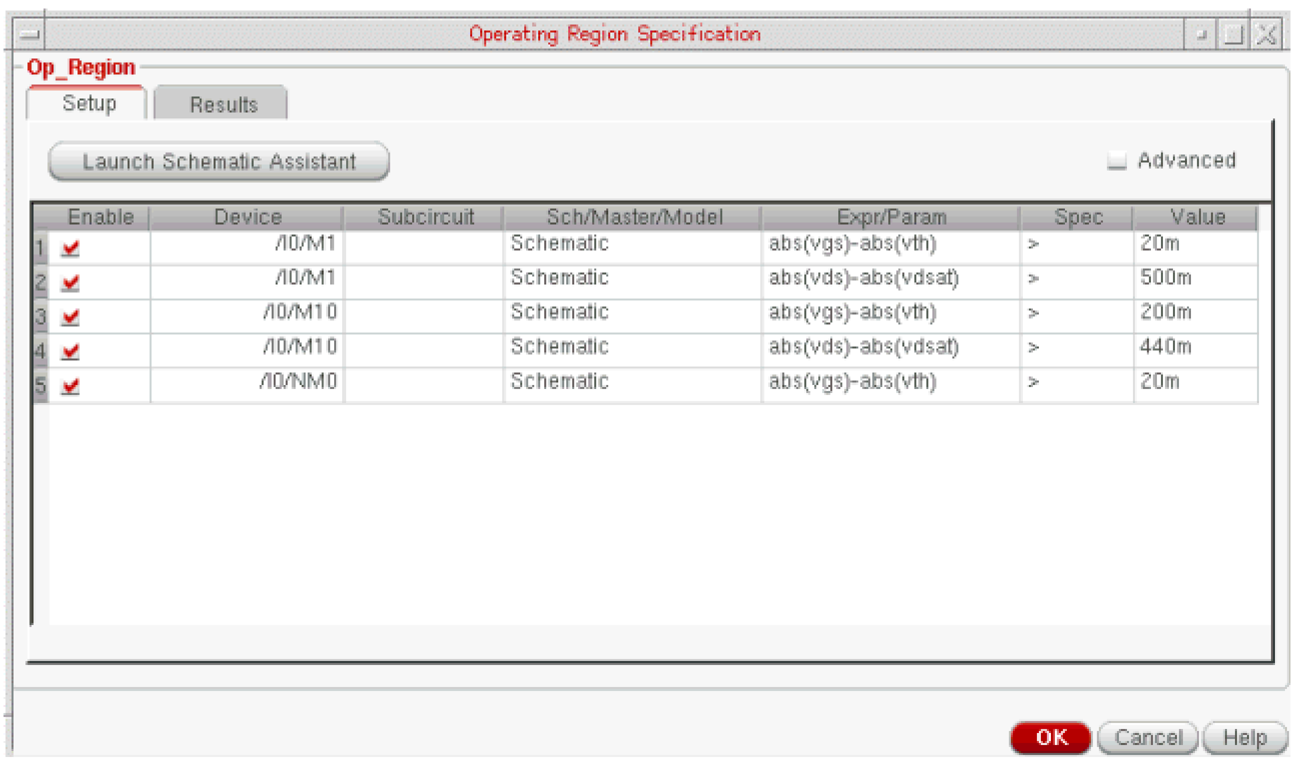
Modifying Operating Region Specifications

To modify an operating region specification, do the following:

1. In the Outputs Setup tab, double-click on the *Expression/Signal/File* field in the row for the operating region specification. An ellipse button appears in this field. Click the ellipse button.



The Operating Region Specification form appears displaying the operating region expressions specified for the test.



You can use this form to perform the following tasks:

Task	Description
View operating region expressions	
Modify operating region expressions	To modify an expression, double-click in the *Expr/Param*, *Spec* or *Value* columns and make the required changes. If you need to use a common expression for multiple parameters, you can modify expressions for their rows together. For more details, see Modifying Multiple Expressions Together on page 731.
Copy and paste operating region expressions	For more information, see Copying and Pasting Operating Region Expressions on page 732
Delete operating region expressions	For more information, see Deleting Operating Region Expressions on page 732
Enable or disable operating region expressions	For more information, see Disabling and Enabling Operating Region Expressions on page 732
Click the *Launch Schematic Assistant* to open the schematic for the test with the Operating Region and Navigator assistant panes docked. You can use the Operating Region assistant pane to quickly add or modify operating region expressions for instances you select on the schematic or in the Navigator assistant pane. **Note:** The Operating Region assistant pane displays only the operating region expressions for the instances that are currently selected on the schematic or in the Navigator assistant pane.	For more information, see Specifying Operating Region Expressions Using the Operating Region Assistant Pane on page 717

Task	Description
Click the *Advanced* check box to access a detailed user interface for adding or modifying operating region expressions	For more information, see Specifying Operating Region Expressions Using the Operating Region Specification Form on page 722

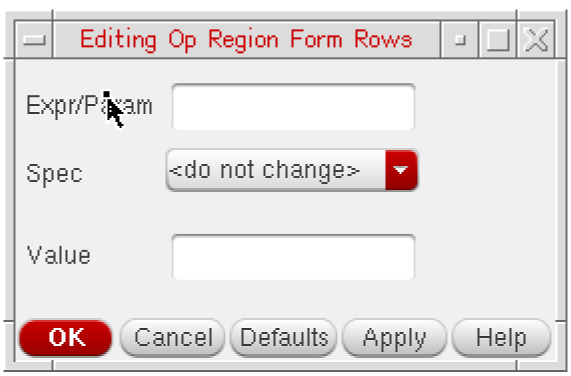
2. Modify the operating region expressions as required.
3. Click *OK*.

Modifying Multiple Expressions Together

To modify multiple operating region expressions together, do the following:

1. Hold down the *Ctrl* key and select more than one expressions.
2. Right-click and choose *Edit Selected Rows*.

   The Editing Op Region Form Rows form appears.



3. Enter an expression in the *Expr/Param* field.
4. Enter a spec value in the *Value* field.
5. Click *OK*.

The expression that you specified in this form is updated in the *Expr/Param* column of all the selected rows in the Operating Region Specifications form.

Copying and Pasting Operating Region Expressions

To copy an operating region expression, do the following:

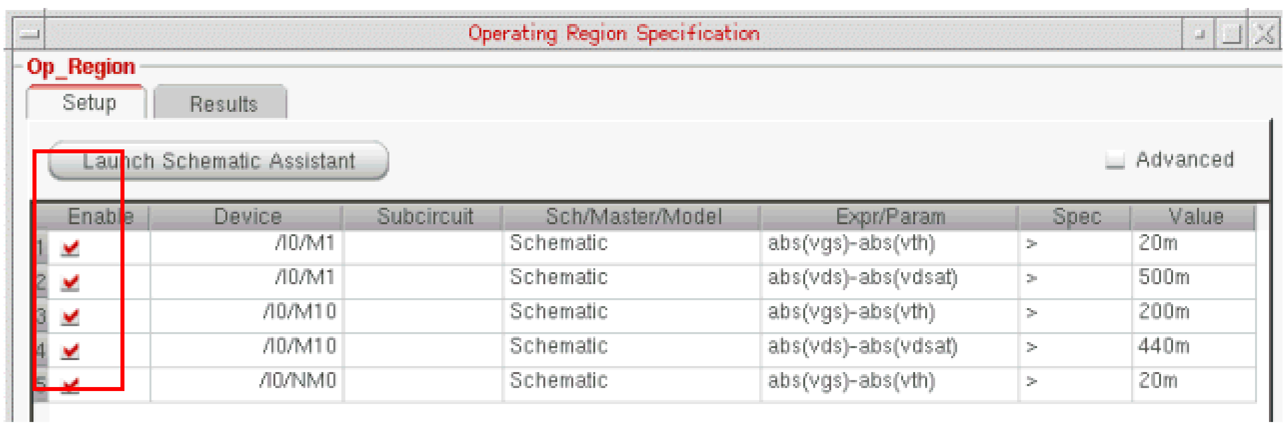
➡ Right-click the row for the expression you want to copy and select *Copy Expression*.

To paste an operating region expression, do the following:

➡ Right-click the row where you want to paste the expression and select *Paste*.

Disabling and Enabling Operating Region Expressions

The *Enable* column in the Operating Region Specification form indicates whether an operating region expression is enabled or disabled. By default, the operating region expressions you add are automatically enabled.



To disable an operating region expression, do the following:

➤ Deselect the check box next to the expression in the *Enable* column.

To enable an operating region expression, do the following:

➤ Select the check box next to the expression in the *Enable* column.

Deleting Operating Region Expressions

To delete an operating region expression, do the following:

➡ Right-click the row for the expression and choose *Delete*.

Saving Operating Region Expressions to a File

You can save the operating region expressions specified in the Operating Region assistant pane or the detailed view of the Operating Region Specification form to a file. You can then load the file to quickly add operating region expressions. For more information about loading operating region expressions, see Loading Operating Region Expressions from a File on page 735.

To save operating region expressions to a file, do the following:

1. On the Operating Region assistant pane or the detailed view of the Operating Region Specification form, in the *Label* field, type a name for the specified set of operating region expressions.

   For example, in the following figure, the two expressions specified in the Operating Region assistant pane will be saved with the name `myOpRegionExpr`.

Figure 15-1 Saving Expressions in Operating Region Assistant Pane

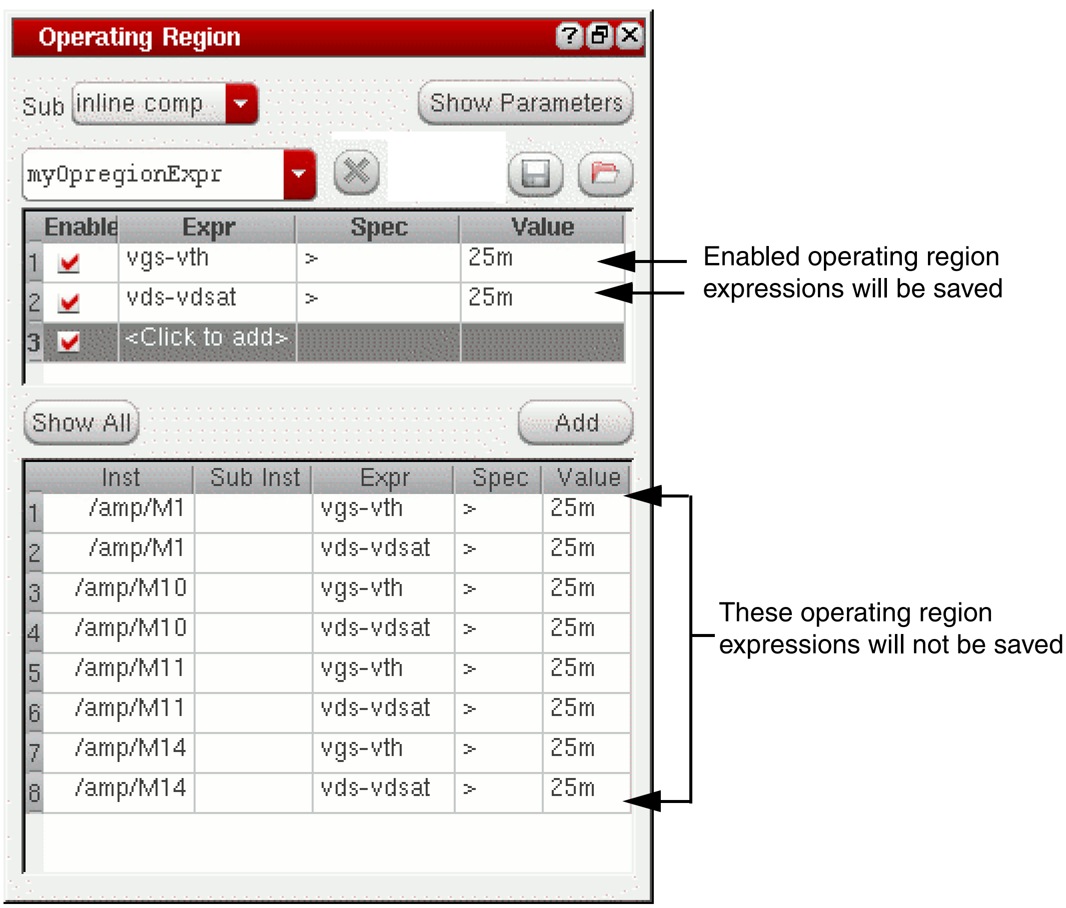
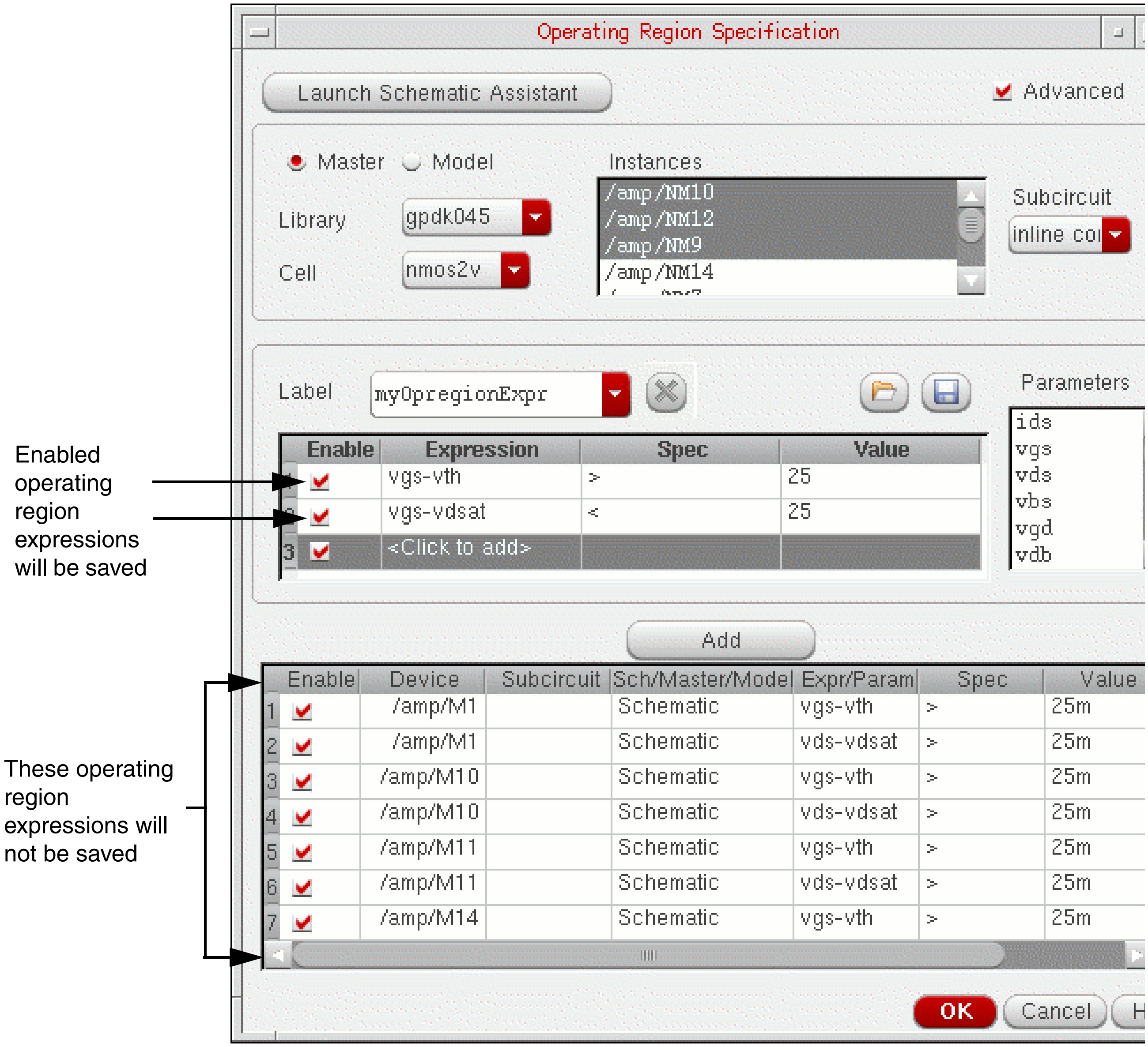


Figure 15-2 Saving Expressions in Operating Region Specification Form



2. Ensure that the expressions you want to save are enabled, and the expressions you don't want to save are disabled.

   ❑ To enable an expression, select the check box next to it in the *Enable* column.

   ❑ To disable an expression, deselect the check box next to it in the *Enable* column.

3. Click the *Save* button.

   The Select Operating Region Expression File form appears.

4. Specify the name and location of the file and click *Save*.

The enabled expressions are saved in the file.

Important

If you specify name of an existing file, the tool prompts you to check if you want to replace the existing file or specify another file name. The tool does not append the expressions to the existing file.

Tip

You can concatenate the expressions specified in multiple files and save in a common file. In this case, when the file is loaded, all expressions saved with a common label name are displayed together.

Loading Operating Region Expressions from a File

You can load operating region expressions from a file into the Operating Region assistant pane and the detailed view of the Operating Region Specification form.

To load operating region expressions from a file, do the following:

1. Click the *Load* button in the Operating Region assistant pane or the detailed view of the Operating Region Specification form.

The Select Operating Region Expression File form appears.

2. Select the file and click *Open*.

The expressions in the file are displayed. The name specified for the set of expressions in the file is also displayed in the *Label* drop-down list.

Tip

The names specified for the sets of expressions in the files loaded during the current ADE GXL session are displayed in the *Label* drop-down list. Therefore, during the current ADE GXL session, you can select a name from the *Label* drop-down list to add expressions, instead of loading the corresponding file again.

Tip

Expressions saved in a tab- or comma-separated file can also be loaded in the Operating Region assistant pane or the Operating Region Specification form. In this case, if the expressions are not associated with a label name, the *Label* list displays `<none>`.

Deleting Operating Region Specifications

To delete an existing operating region specification, do the following:

- ➡ In the Outputs Setup tab, right-click the row for the operating region specification and choose *Delete Output*.
- ➡ In the Operating Region assistant pane or the detailed view of the Operating Region Specification form, select a label from the *Label* list and click *Delete*. All the expressions saved with the selected label name are deleted.

Disabling and Enabling Operating Region Specifications

To disable an operating region specification, do the following:

- ➡ In the *Plot* column on the Outputs Setup tab, deselect the check box next to the specification.
- ➡ Right-click on an operating region specification and choose *Disable Plot*.

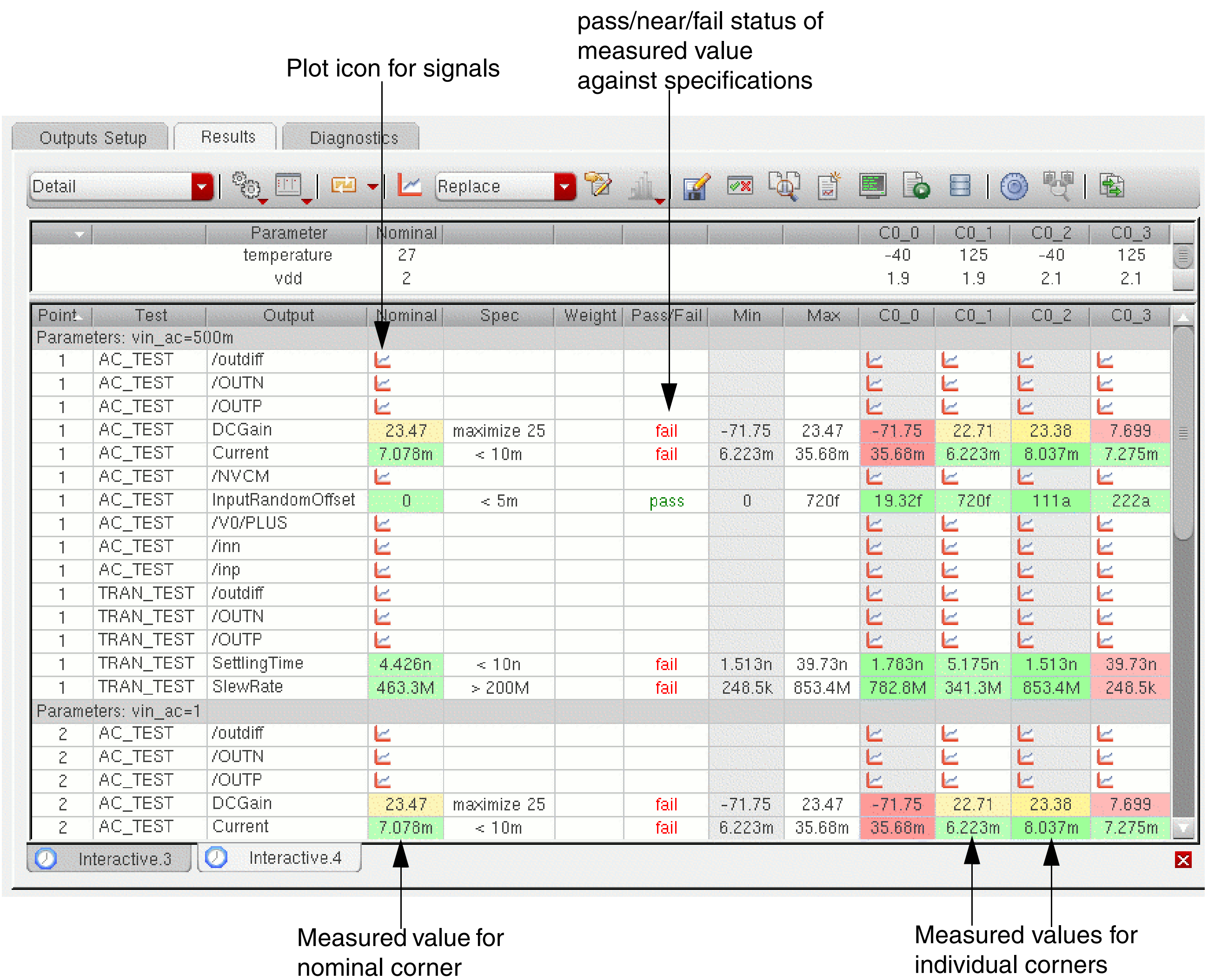
To enable an operating region specification, do the following:

- ➡ In the *Plot* column on the Outputs Setup tab, select the check box next to the specification.
- ➡ Right-click on an operating region specification and choose *Enable Plot*.

Viewing Specification Results in the Results Tab

When you run a simulation, the simulation results are displayed in the Results tab of the Outputs pane. For more information about running simulations, see Chapter 13, "Running Simulations."

Figure 15-3 Display of Simulation Results in the Results Tab



Parameter	Nominal	C0_0	C0_1	C0_2	C0_3
temperature	27	-40	125	-40	125
vdd	2	1.9	1.9	2.1	2.1

Point	Test	Output	Nominal	Spec	Weight	Pass/Fail	Min	Max	C0_0	C0_1	C0_2	C0_3
Parameters: vin_ac=500m												
1	AC_TEST	/outdiff										
1	AC_TEST	/OUTN										
1	AC_TEST	/OUTP										
1	AC_TEST	DCGain	23.47	maximize 25		fail	-71.75	23.47	-71.75	22.71	23.38	7.699
1	AC_TEST	Current	7.078m	< 10m		fail	6.223m	35.68m	35.68m	6.223m	8.037m	7.275m
1	AC_TEST	/NVCM										
1	AC_TEST	InputRandomOffset	0	< 5m		pass	0	720f	19.32f	720f	111a	222a
1	AC_TEST	/V0/PLUS										
1	AC_TEST	/inn										
1	AC_TEST	/inp										
1	TRAN_TEST	/outdiff										
1	TRAN_TEST	/OUTN										
1	TRAN_TEST	/OUTP										
1	TRAN_TEST	SettlingTime	4.426n	< 10n		fail	1.513n	39.73n	1.783n	5.175n	1.513n	39.73n
1	TRAN_TEST	SlewRate	463.3M	> 200M		fail	248.5k	853.4M	782.8M	341.3M	853.4M	248.5k
Parameters: vin_ac=1												
2	AC_TEST	/outdiff										
2	AC_TEST	/OUTN										
2	AC_TEST	/OUTP										
2	AC_TEST	DCGain	23.47	maximize 25		fail	-71.75	23.47	-71.75	22.71	23.38	7.699
2	AC_TEST	Current	7.078m	< 10m		fail	6.223m	35.68m	35.68m	6.223m	8.037m	7.275m

Tip

After running a simulation, if you add or modify measurement expressions or specifications in the Outputs Setup tab, you can click the button on the Results tab to update the results displayed in the Results tab based on the new or modified expressions and specifications. For more information about using the button, see Re-evaluating Expressions and Specifications on page 650.

The results will also be updated based on the new or modified expressions and specifications if you click the button on the Results tab to plot across all points. For more information about using the button, see Plotting Results on page 624.

The Results tab displays a plot icon for the signals that you have selected for plotting or saving on the Outputs Setup tab of the Outputs pane. For more information about selecting signals for plotting or saving, see Specifying Whether a Result Will Be Saved or Plotted on page 375.

For measurements defined in the Outputs Setup tab, the program measures the result against the performance specification and displays the following information for each simulation in the Results tab.

- Measured value for the nominal corner in the *Nominal* column.
- Measured value for each corner in the column for the corner.
- *pass*, *near* or *fail* status for the measured value from each simulation in the *Pass/Fail* column.
    - *pass* means that all the measured values are within the limits defined by the specification.

      For example, in Figure 15-3, the measured value `506.7m` for the `Gain_commonMode` measurement has the *pass* status because it falls within the target value `<550m` for the measurement.
    - *near* means that one or more measured values are no more than 10% outside the target value of the specification.

      For example, in Figure 15-3, the measured value `94.24` for the `Gain_openLoop` measurement has the *near* status because it falls near (within 10% of) the target value `>95` for the measurement.
    - *fail* means that one or more measured values are greater than 10% outside the target value of the specification.

For example, in Figure 15-3, the measured value `118.4` for the `AC_gain_1KHz` measurement has the *fail* status because it falls outside the target value `>130` for the measurement.

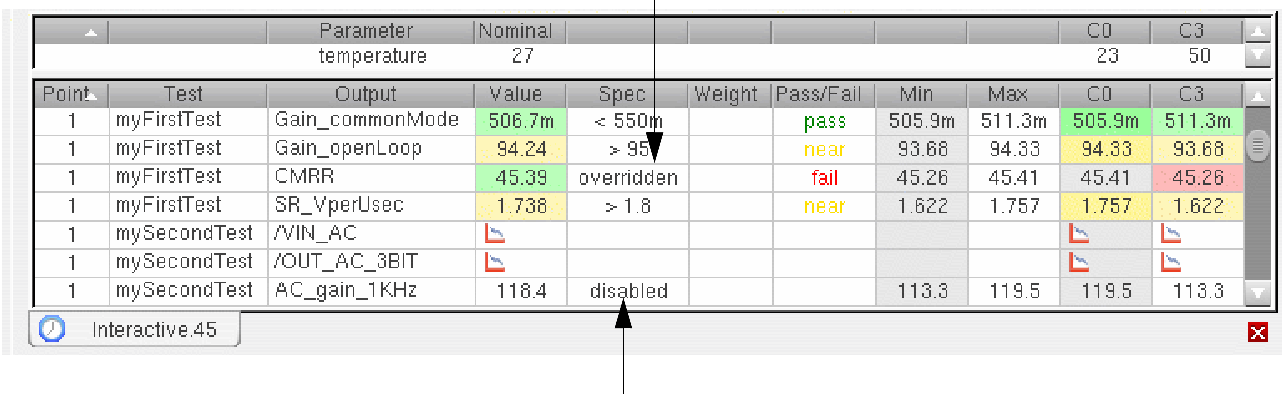
The measured values with a *pass* status are displayed with a green background, those with a *near* status appear with a yellow background, and those with a *fail* status appear with a red background.

Note the following:

- ❑ The *Pass/Fail* column displays the *pass* status for a measurement only if the measured values for all the corners for the measurement have a pass status (displayed with a green background). For example, in 15-3, the `Gain_commonMode` measurement has the *pass* status because the measured values for the nominal corner (displayed in the *Nominal* column) and the corners *C0* and *C3* have a pass status.
- ❑ If the measured value for any corner has a *near* status (displayed with a yellow background), the *Pass/Fail* column displays the *near* status for the measurement.
- ❑ If the measured value for any corner has a fail status (displayed with a red background), the *Pass/Fail* column displays the *fail* status for the measurement.

■ The text `overridden` displayed next to a measurement in the *Spec* column on the Results tab indicates that you have overridden or disabled one or more corner specifications for the measurement. For example, in Figure 15-4, the text `overridden` next to the `CMRR` measurement in the *Spec* column indicates that you have overridden or disabled one or more corner specifications for the `CMRR` measurement.

Figure 15-4 Display in Results Tab when Corner Specs are Disabled or Overridden



		Parameter	Nominal						C0	C3
		temperature	27						23	50
Point	**Test**	**Output**	**Value**	**Spec**	**Weight**	**Pass/Fail**	**Min**	**Max**	**C0**	**C3**
1	myFirstTest	Gain_commonMode	506.7m	< 550m		pass	505.9m	511.3m	505.9m	511.3m
1	myFirstTest	Gain_openLoop	94.24	> 95		near	93.68	94.33	94.33	93.68
1	myFirstTest	CMRR	45.39	overridden		fail	45.26	45.41	45.41	45.26
1	myFirstTest	SR_VperUsec	1.738	> 1.8		near	1.622	1.757	1.757	1.622
1	mySecondTest	/VIN_AC								
1	mySecondTest	/OUT_AC_3BIT								
1	mySecondTest	AC_gain_1KHz	118.4	disabled			113.3	119.5	119.5	113.3

Do the following to view the specification for the measurement and for each corner:

- ❑ To view the specification for the measurement, hover the mouse pointer over the text `overridden` in the *Spec* column.

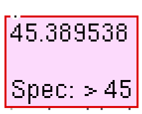
  The specification for the measurement is displayed in a pop-up, as shown below:

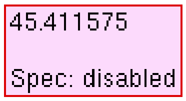
- ❑ To view the specification for the nominal corner, hover the mouse pointer over the measured value for the nominal corner that is displayed next to the measurement in the *Nominal* column.

  To view the specification for a corner, hover the mouse pointer over the measured value for the corner that is displayed next to the measurement in the column for the corner.

  The measured value and specification for the corner is displayed in a pop-up, as shown below:

If a corner specification is disabled for the measurement, the pop-up displays the measured value and disabled status for the corner as shown below:

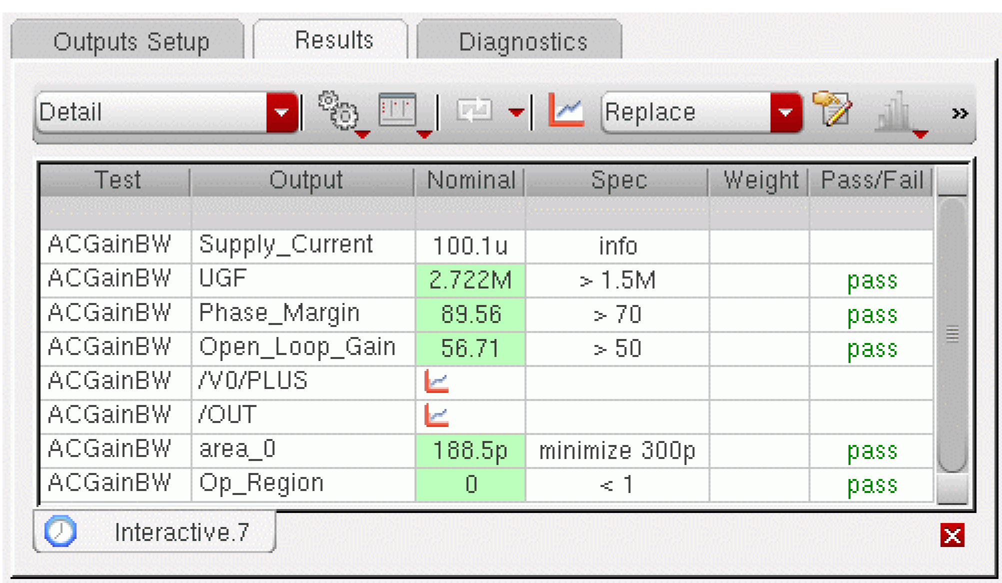


- The text `disabled` displayed next to a measurement in the *Spec* column on the Results tab indicates that you have disabled all the corner specifications for the measurement. For example, in Figure 15-4, the text `disabled` next to the `AC_gain_1KHz` measurement in the *Spec* column indicates that you have disabled all the corner specifications for the `AC_gain_1KHz` measurement.

Viewing Operating Region Violations

When you run a simulation, ADE GXL combines the individual device expressions in the operating region specification into a single output and calculates the total error. The total error is the sum of all the individual violation margins plus an offset of 1.

An operating region specification is met if the results for all its operating region expressions fall within their corresponding target values. The Results tab displays a value of `0` and a `pass` status if an operating region specification is met. For example, in the following figure, the operating region specification for the `ACGainBW` test has a `pass` status because its value is `0`.

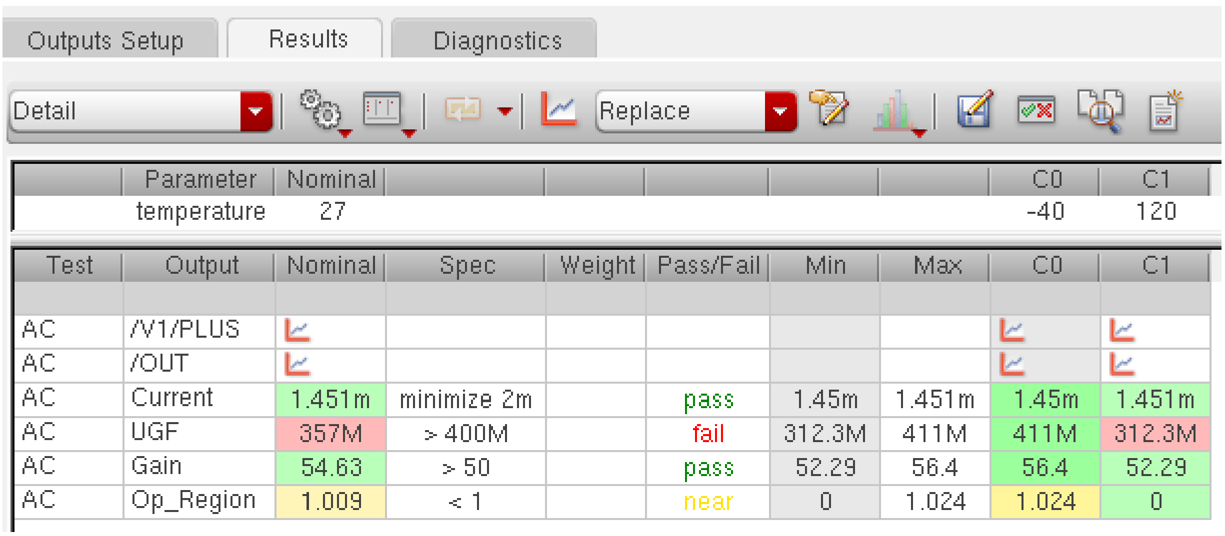


A violation occurs when the result for any operating region expression in the specification falls outside its specified target value. In this case, the Results tab displays a value greater than `0` that is calculated as, shown below:

failed operating region result = 1 + sum of fail margin for failed expressions

Note: The operating region result is never between `0` and `1`.

A `near` or `fail` status depending on the range of deviation from the target spec value. For example, in the following figure, the operating region specification for the `AC` test, `Op_Region`, has a `near` status because its value is slightly greater than `1` whereas the spec requires the value to be less than 1.



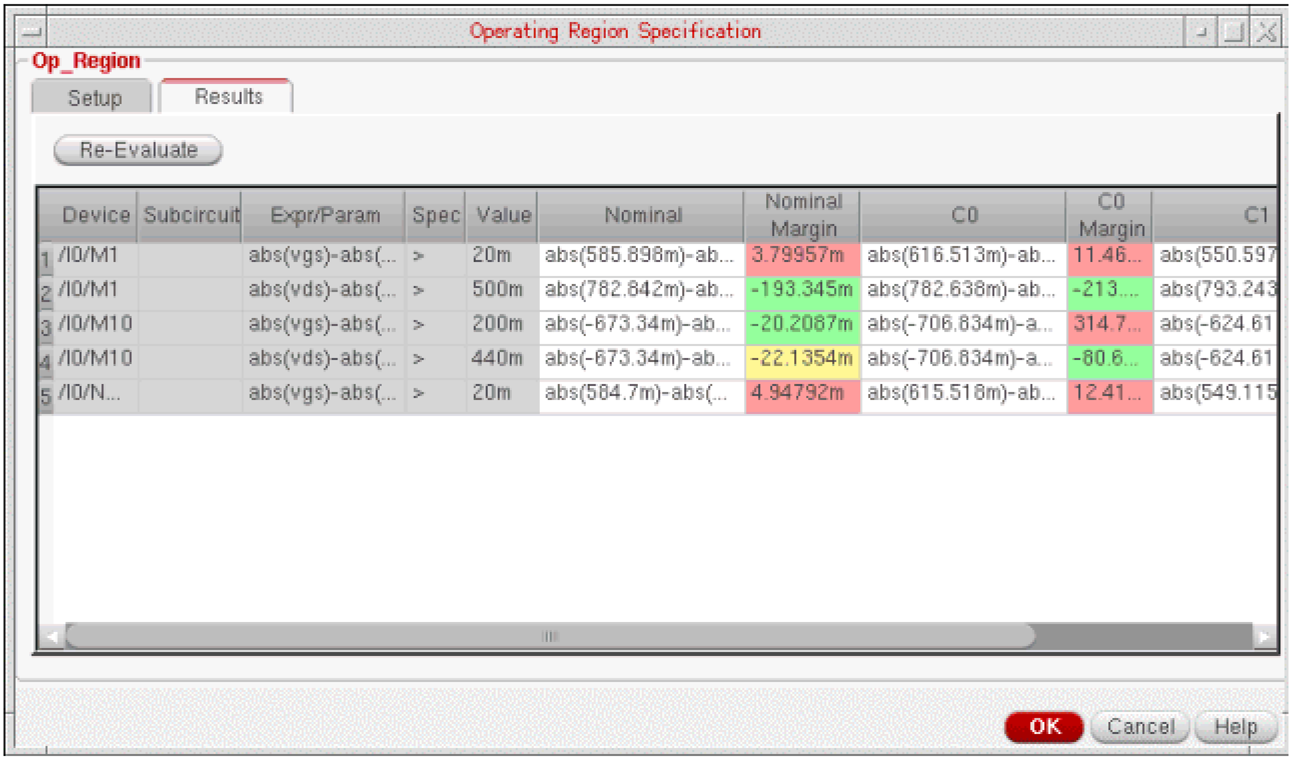
	Parameter	Nominal						C0	C1
	temperature	27						-40	120

Test	Output	Nominal	Spec	Weight	Pass/Fail	Min	Max	C0	C1
AC	/V1/PLUS								
AC	/OUT								
AC	Current	1.451m	minimize 2m		pass	1.45m	1.451m	1.45m	1.451m
AC	UGF	357M	> 400M		fail	312.3M	411M	411M	312.3M
AC	Gain	54.63	> 50		pass	52.29	56.4	56.4	52.29
AC	Op_Region	1.009	< 1		near	0	1.024	1.024	0

To view operating region violations, do the following:

➡ In the Results tab of the Outputs pane, right-click the operating region specification and choose *Op. Region Violations*.

The Operating Region Specification form is displayed with the *Results* tab in view. This tab displays the results of all operating region expressions in a tabular format, as shown below:



Two columns are shown for each corner—the column with corner name shows the result of expression evaluation for that corner and the *Margin* column next to it shows the margin by which the result deviates from the spec value. Depending on the difference between the spec and margin values, cells in the *Margin* column are colored as green, red, and yellow to highlight the `pass`, `fail`, or `near` status, respectively.

For more information about using the Results tab of the Operating Region Specification form, see the following sections:

- Re-evaluating Operating Region Expressions on page 744
- Highlighting Operating Region Violations on the Schematic on page 744
- Unhighlighting Operating Region Violations on the Schematic on page 745
- Hiding and Showing a Single Column on page 746
- Hiding and Showing Multiple Columns on page 746
- Hiding and Showing Only Expression Columns or Margin Columns on page 747

Re-evaluating Operating Region Expressions

Important

Make sure that you load the setup to active before re-evaluating outputs on the Outputs Setup tab. You cannot re-evaluate operating region expressions only by viewing results.

If required, you can edit the expressions that do not meet the specs and re-evaluate results. For this, do the following in the Operating Region Specification form:

1. Open the *Setup* tab and add, edit, or delete expressions, as required.
2. Open the *Results* tab and click *Re-evaluate*.

The results for all the expressions are re-evaluated for all the corners and updated on the *Results* tab. If you get the desired results, click *OK* to save the changes in the operating region specification. Now, to see the result of change in operating region specification on the operating region output, click  on the *Results* tab of ADE XL Outputs pane. All the expressions and operating region outputs are re-evaluated. The operating region output is now the result of new operating region specifications.

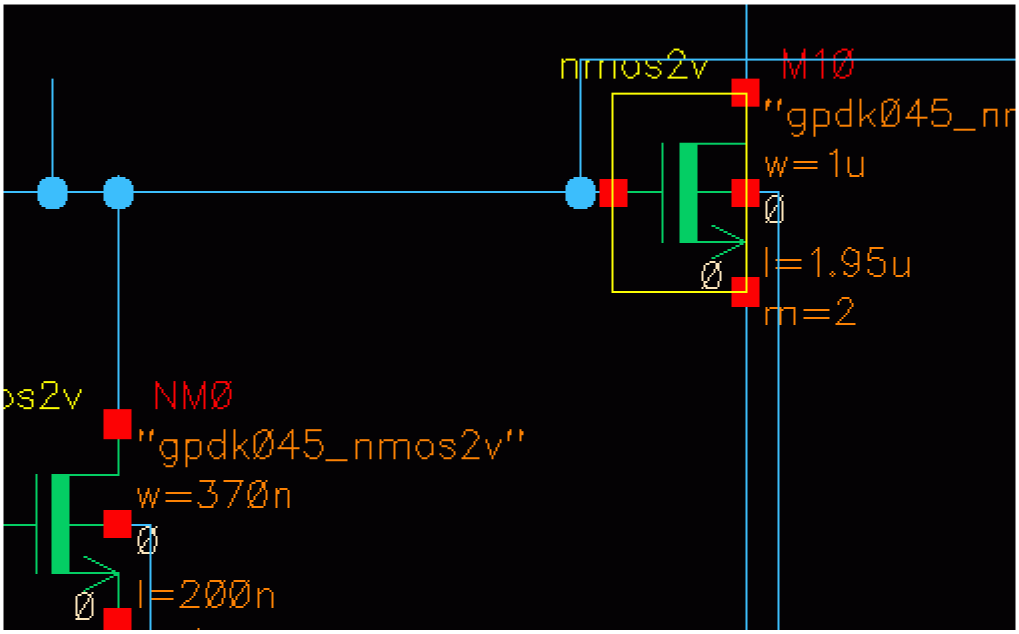
Highlighting Operating Region Violations on the Schematic

You can highlight the instance on the schematic for which an operating region expression has failed. This helps you to quickly correct the design or the operating region expression.

To highlight the instance for which an operating region expression has failed, do the following:

1. Right-click on the row for which an expression has failed.
2. Choose *Highlight*.

The instance for which the expression failed is highlighted in yellow on the schematic. For example, in the following figure, the instance `M10` for which an expression failed is highlighted on the schematic.



Tip

To highlight the instances corresponding to multiple failed expressions, hold down the *Shift* key (for contiguous selection) or the *Ctrl key* (for noncontiguous selection) and click the row for the next failed expression to add more expressions to the selection set, then click the *Highlight* button.

Note: In a hierarchical design, a block is highlighted in yellow if an operating region expression has failed for an instance in the block. When you descend into the block, the instance for which the expression has failed is also highlighted in yellow.

Unhighlighting Operating Region Violations on the Schematic

To clear the highlight from all the instances corresponding to the operating region violations on the schematic, do the following:

1. Right-click on the row for which an expression has failed.
2. Choose *Clear*.

Hiding and Showing a Single Column

You can hide and show the required columns in the *Setup* or *Results* tab of the Operating Region Specification form.

To hide a column that is currently displayed, do the following:

1. Right-click on a column name to display a pop-up menu.

   A tick mark next to a column name in the pop-up menu indicates that the column is currently displayed.

2. Choose the name of the column you want to hide.

To show a column that is currently not displayed,

➡ Right-click on a column name and choose the name of the column you want to display from the pop-up menu.

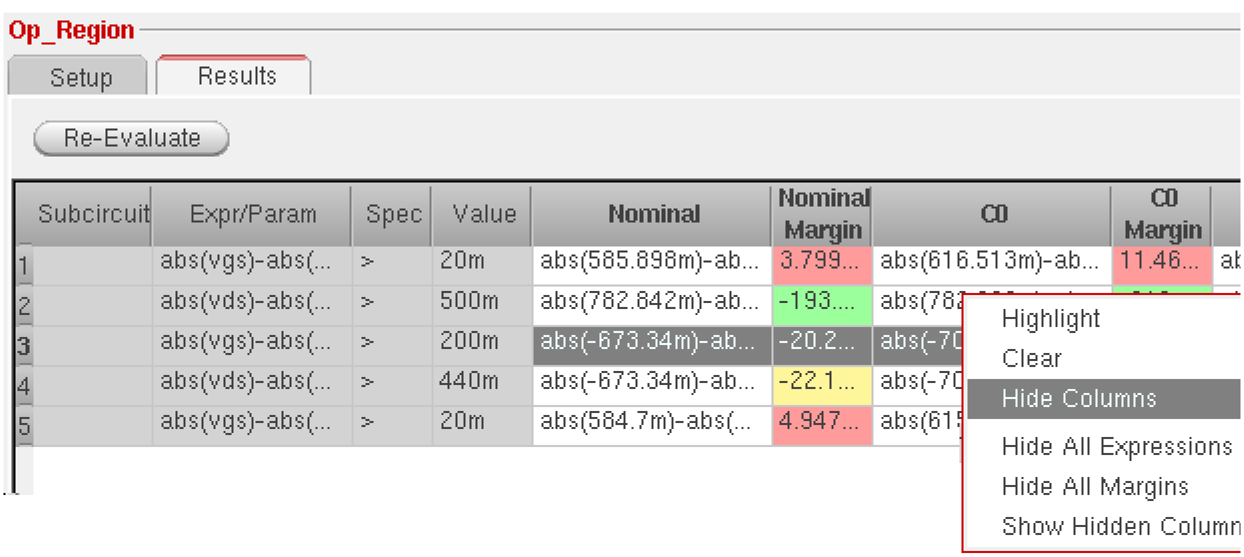
Hiding and Showing Multiple Columns

You can hide and show the required columns in the *Setup* or *Results* tab of the Operating Region Specification form.

To hide multiple columns that are currently displayed, do the following:

1. Hold down the Ctrl key and click in any cell in the columns that you want to hide.

2. Right-click in any of the selected cells and choose *Hide Columns*.



All the selected columns are hidden.

To display all the columns that are currently hidden,

➡ Right-click in any cell and choose *Show Hidden Columns*.

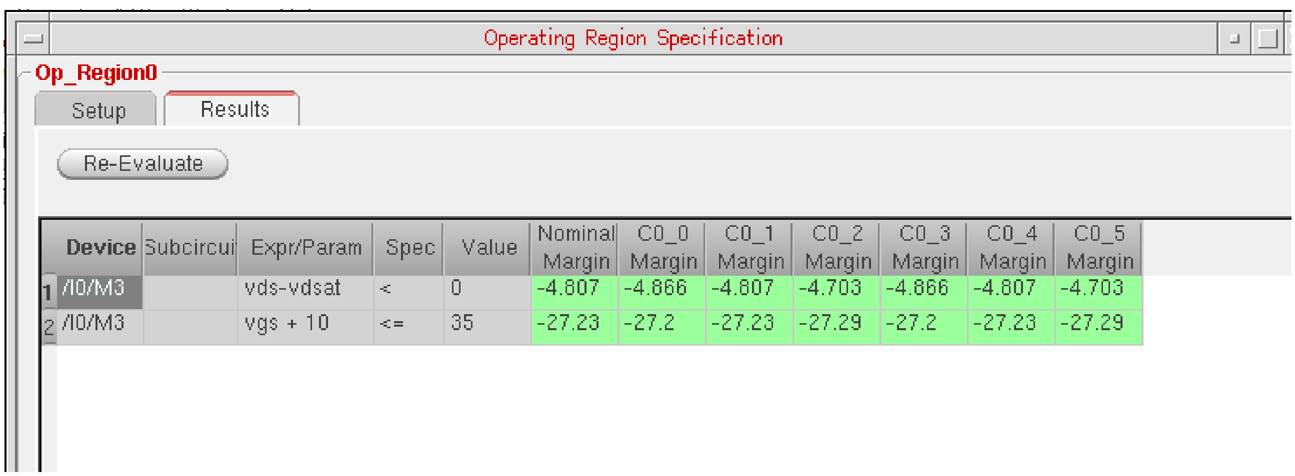
All the hidden columns become visible on the Results tab.

Hiding and Showing Only Expression Columns or Margin Columns

If multiple columns are displayed on the *Results* tab of the Operating Region Specification form, you can show only expression columns or margin columns. This helps in showing only one type of results at a time.

To hide all expression columns and to show only the margin columns:

➡ Right-click in any of the results cells and choose *Hide All Expressions*. All expression columns are hidden and only the margin columns appear, as shown below.



	Device	Subcircui	Expr/Param	Spec	Value	Nominal Margin	C0_0 Margin	C0_1 Margin	C0_2 Margin	C0_3 Margin	C0_4 Margin	C0_5 Margin
1	/I0/M3		vds-vdsat	<	0	-4.807	-4.866	-4.807	-4.703	-4.866	-4.807	-4.703
2	/I0/M3		vgs + 10	<=	35	-27.23	-27.2	-27.23	-27.29	-27.2	-27.23	-27.29

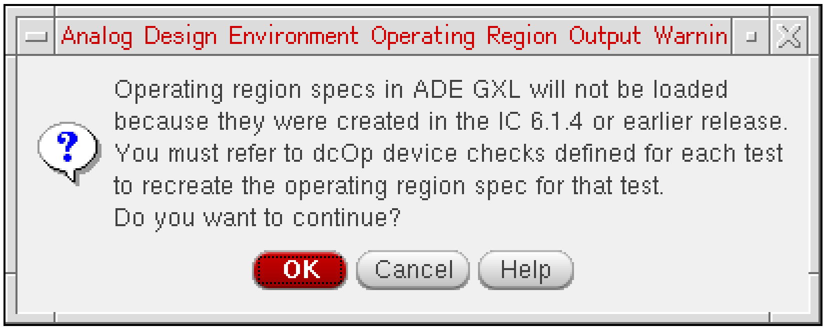
Tip

An alternate way to hide multiple columns is to right-click on any column header and clear the check box for columns to be hidden. However, the *Hide All Margins* or *Hide All Expressions* commands provide quick options to hide multiple columns.

To unhide the hidden margin columns, right-click in any cell and choose *Show Hidden Columns*.

Migrating Operating Region Specifications from IC6.1.4 to IC6.1.5

Starting from IC6.1.5, ADE XL provides a new user interface for defining operating region expressions. Because of this, the following message is displayed if an ADE XL view that contains operating region specifications created in the IC6.1.4 or earlier release is opened in the IC6.1.5 or later release.



Do the following to migrate operating region expressions created in IC6.1.4 or earlier releases to the IC6.1.5 format:

1. Click *OK* to open the ADE XL view.

   The operating region expressions defined for each test's operating region specification in the view are displayed as device checks for the test in the Outputs Setup tab.

2. In the Outputs Setup tab, double-click on the *Expression/Signal/File field* in the row for a device check.

   The Device Check Specification form appears displaying the operating region expression details.

3. Define the operating region expressions for the test using the procedure described in Setting Up Operating Region Specifications on page 712.

4. Delete the device checks that have been defined as operating region expressions.

For more information about device checks, see the Device Checking chapter.

Working with the Specification Summary

The Spec Summary form provides a mechanism for managing and organizing simulation results for different purposes:

- Create a spec summary for design reviews
- Generate a summary report of pass/fail results for measurements
- Display results for quick inspection of multiple simulation runs
- Verify a design against specifications
- Export the spec summary to HTML or CSV format files for printing purposes.

For more information, see the following topics:

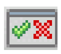
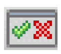
- Viewing the Spec Summary on page 750
- Saving a Spec Summary on page 753
- Opening a Spec Summary on page 754
- Adding a Specification to the Spec Summary on page 754
- Deleting a Specification from the Spec Summary on page 755
- Changing the History Item from which Results are Displayed on page 755
- Updating the Spec Summary with the Latest Results on page 756
- Recreating the Spec Summary from the Results in the Active Results Tab on page 757
- Viewing the Detailed Results for Specifications on page 757
- Plotting the Results for Specifications on page 759
- Exporting a Spec Summary to a HTML or CSV File on page 759
- Sorting Data in the Spec Summary Form on page 760
- Hiding and Showing Columns in the Spec Summary Form on page 760

Viewing the Spec Summary

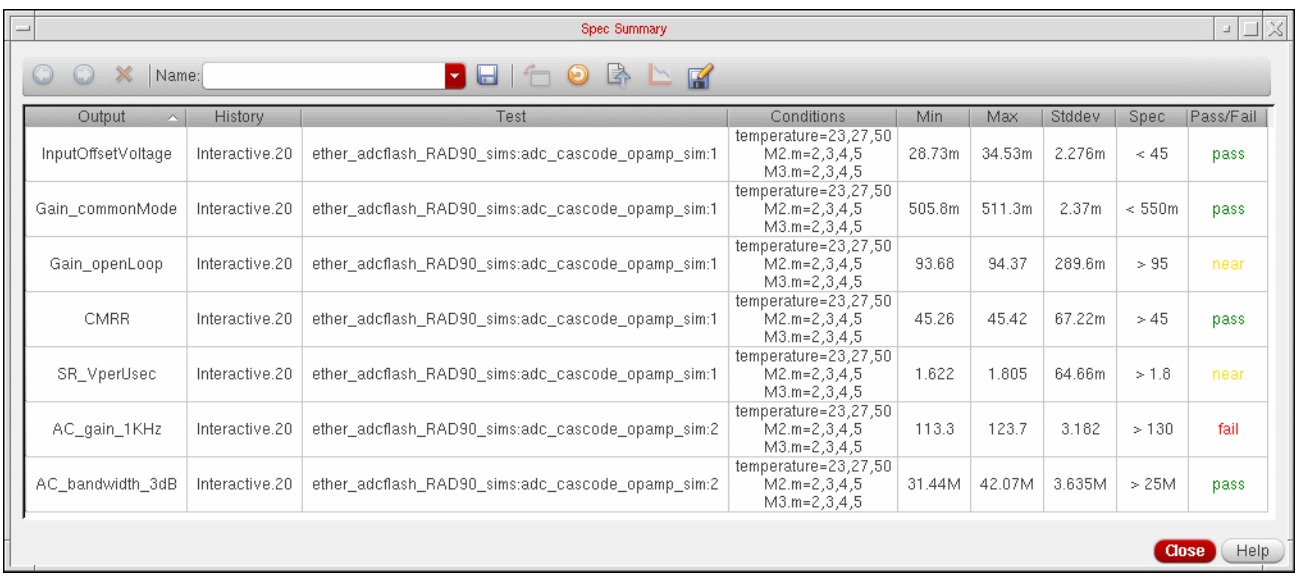
To view the specification (spec) summary, do the following on the Results tab of the Outputs pane:

Note: You can view the spec summary only for the results of a Single Run, Sweeps and Corners or Monte Carlo Sampling simulation run. You cannot view the spec summary for the results of a Global Optimization, Local Optimization, , Feasibility Analysis, Size Over Corners or Sensitivity Analysis simulation run.

➤ Do one of the following on the currently active Results tab for a Single Run, Sweeps and Corners or Monte Carlo simulation run:

- ❑ Choose *Create – Spec Summary*.
- ❑ Click the  toolbar button on the ADE XL window.
- ❑ Click the  button on the Results tab.

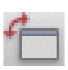
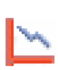
The spec summary for the results in the currently active Results tab is displayed in the Spec Summary form.



Output	History	Test	Conditions	Min	Max	Stddev	Spec	Pass/Fail
InputOffsetVoltage	Interactive.20	ether_adcflash_RAD90_sims:adc_cascode_opamp_sim:1	temperature=23,27,50 M2.m=2,3,4,5 M3.m=2,3,4,5	28.73m	34.53m	2.276m	< 45	pass
Gain_commonMode	Interactive.20	ether_adcflash_RAD90_sims:adc_cascode_opamp_sim:1	temperature=23,27,50 M2.m=2,3,4,5 M3.m=2,3,4,5	505.8m	511.3m	2.37m	< 550m	pass
Gain_openLoop	Interactive.20	ether_adcflash_RAD90_sims:adc_cascode_opamp_sim:1	temperature=23,27,50 M2.m=2,3,4,5 M3.m=2,3,4,5	93.68	94.37	289.6m	> 95	near
CMRR	Interactive.20	ether_adcflash_RAD90_sims:adc_cascode_opamp_sim:1	temperature=23,27,50 M2.m=2,3,4,5 M3.m=2,3,4,5	45.26	45.42	67.22m	> 45	pass
SR_VperUsec	Interactive.20	ether_adcflash_RAD90_sims:adc_cascode_opamp_sim:1	temperature=23,27,50 M2.m=2,3,4,5 M3.m=2,3,4,5	1.622	1.805	64.66m	> 1.8	near
AC_gain_1KHz	Interactive.20	ether_adcflash_RAD90_sims:adc_cascode_opamp_sim:2	temperature=23,27,50 M2.m=2,3,4,5 M3.m=2,3,4,5	113.3	123.7	3.182	> 130	fail
AC_bandwidth_3dB	Interactive.20	ether_adcflash_RAD90_sims:adc_cascode_opamp_sim:2	temperature=23,27,50 M2.m=2,3,4,5 M3.m=2,3,4,5	31.44M	42.07M	3.635M	> 25M	pass

The toolbar in the Spec Summary form is described in the following table:

Table 15-3 Spec Summary Toolbar

Icon	Name	Description
	Back to Summary View	Switches from the detail view to the summary view. For more information, see Viewing the Detailed Results for Specifications on page 757.
	Show Detail View	Displays the detailed results for the selected specifications in the detail view. For more information, see Viewing the Detailed Results for Specifications on page 757.
	Delete	Deletes the selected specifications. For more information, see Deleting a Specification from the Spec Summary on page 755.
	Save Spec Summary	Saves the spec summary. For more information, see Saving a Spec Summary on page 753.
	Transpose Table	Displays the results in the detail view in horizontal or vertical format. For more information, see Viewing the Detailed Results for Specifications on page 757.
	Update with Latest Results Data	Updates the spec summary with the latest results from the history item selected for each specification. For more information, see Updating the Spec Summary with the Latest Results on page 756.
	Recreate for the Current History Results	Recreates the spec summary using the results in the active Results tab. For more information, see Recreating the Spec Summary from the Results in the Active Results Tab on page 757.
	Plot Signals	Plots the results for the selected row. For more information, see Plotting the Results for Specifications on page 759.

Icon	Name	Description
	Export to CSV or HTML File	Exports a spec summary to a HTML or comma-separated values (CSV) file For more information, see Exporting a Spec Summary to a HTML or CSV File on page 759.

The columns in the Spec Summary form are described in the following table:

Table 15-4 Spec Summary Columns

Column	Description
Output	Displays the names of output expressions for which the specifications are evaluated.
History	Displays the history items from which the results of the specifications for the output expressions are displayed.
Test	Lists the tests used to generate the results for the output expressions.
Conditions	Displays the values of each swept parameter as a list or range of values. For example, if a parameter `p` is swept through a list of values `x`, `y`, and `z`, then the *Conditions* column displays the values as `p=x,y,z` If a parameter `p` is swept through a range of values, the *Conditions* column displays the values as *`p=startValue`*`:`*`increment`*`:`*`stopValue`*
Min	Displays the minimum measured values for all simulations for the output expressions.
Max	Displays the maximum measured values for all simulations for the output expressions.

Column	Description
Stddev	Displays the standard deviation of the measured values for all simulations of the output expressions. You can hover the mouse cursor over a standard deviation value to view the following information in a pop-up: ■ The number of simulations for the output expression. ■ The mean of the measured values for all simulations for the output expression. ■ The variance in the measured values. **Note:** This is a population standard deviation.
Spec	Displays the specifications for the output expressions.
Pass/Fail	Displays a *pass*, *near* or *fail* status for each specification. ■ *pass* means that all the measured values are within the limits defined by the specification. ■ *near* means that one or more measured values are no more than 10% outside the target value of the specification. ■ *fail* means that one or more measured values are greater than 10% outside the target value of the specification. To investigate the measured values further, see Viewing the Detailed Results for Specifications on page 757

Saving a Spec Summary

To save a spec summary in the Spec Summary form, do the following:

1. In the *Name* field, type a name for the spec summary.
2. Click the  button.

The spec summary is saved to the `documents` folder of the ADE XL view and displayed in the *Documents* tree on the Data View pane. For more information about working with documents, see Chapter 18, “Working with Documents.”

Note: The spec summary file is saved with the `.specsummary` extension.

Opening a Spec Summary

To open an existing spec summary, do one of the following:

- In the *Name* drop-down list on the Spec Summary form, select the spec summary.
- In the *Documents* tree on the Data View pane, double-click on the spec summary.

The spec summary is displayed in the Spec Summary form.

Deleting a Spec Comparison

To delete an existing spec summary, do the following:

➡ In the *Documents* tree on the Data View pane, right-click the spec summary you want to delete and choose *Delete*.

The spec summary will not be displayed in the *Name* drop-down list in the Spec Summary form.

Adding a Specification to the Spec Summary

You can add a specification from the results in the active Results tab or from the results of a history item.

Note: You can add specifications only from the results of a Single Run, Sweeps and Corners or Monte Carlo simulation run. You cannot add specifications from the results of a Global Optimization, Local Optimization, , Feasibility Analysis, Size Over Corners or Sensitivity Analysis simulation run.

To add a specification from the results in the active Results tab for a Single Run, Sweeps and Corners or Monte Carlo simulation run, do the following:

➤ Right-click on a row in the Results tab of the Outputs pane and choose *Add to Spec Summary*.

The specification is added to the spec summary.

Note: You cannot add an output of type signal to the spec summary.

To add a specification from the results of a history item for a Single Run, Sweeps and Corners or Monte Carlo simulation run, do the following:

1. Display the results for the history item. For more information, see Viewing Results from a Particular Checkpoint on page 816.

2. On the Results tab for the history item, right-click a specification and choose *Add to Spec Summary*.

Tip

To add multiple specifications, hold down the *Shift* key (for contiguous selection) or the *Ctrl* key (for noncontiguous selection), right-click and choose *Add to Spec Summary*.

Deleting a Specification from the Spec Summary

To delete a specification from the spec summary, do the following.

➤ Select the specification and click the  button.

To delete multiple specifications, hold down the *Shift* key (for contiguous selection) or the *Ctrl* key (for noncontiguous selection) and click the  button.

Changing the History Item from which Results are Displayed

The *History* column in the Spec Summary form displays the names of the history items from which the results of the specifications are displayed. You can change the history item for a specification to display the results from that history item.

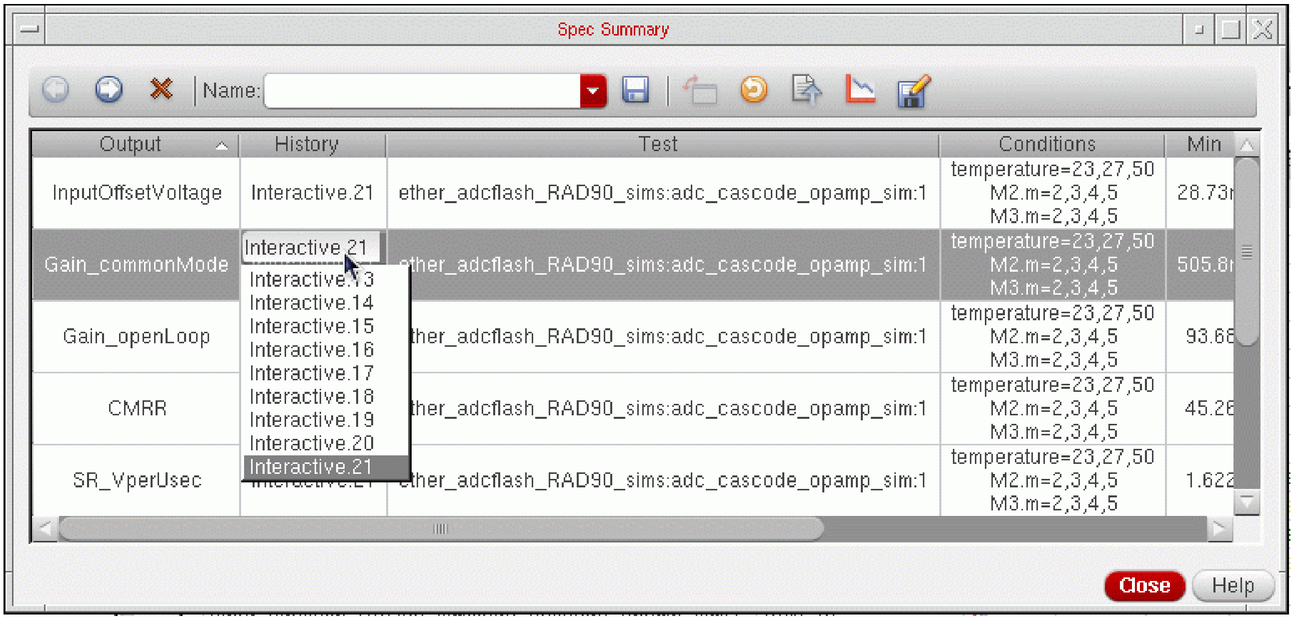
Note: You can change only to a history item for a Single Run, Sweeps and Corners or Monte Carlo simulation run. You cannot change to a history item for a Global Optimization, Local Optimization, , Feasibility Analysis, Size Over Corners or Sensitivity Analysis simulation run.

To change the history item for a single specification, do the following:

1. In the *History* column, click on the name of the history item for the specification.

   A button with the name of the history item appears.

2. Click on the button and select a different history item.



The results for the specification from the selected history item are displayed in the spec summary.

To change the history item for multiple specifications, do the following:

1. Hold down the *Shift* key (for contiguous selection) or the *Ctrl* key (for noncontiguous selection) and click on the name of a history item for any specification in the selection.

   A button with the name of the history item appears.

2. Click on the button and select a different history item.

   The results for the specifications from the selected history item are displayed in the spec summary.

Updating the Spec Summary with the Latest Results

To update the spec summary with the latest results from the history item selected for each specification, do the following:

➤ Click the  button.

Note: If the history item selected for a specification does not exist, the row for the specification is highlighted in red color. You can select a different history item for the specification (see Changing the History Item from which Results are Displayed on page 755)

or delete the specification from the spec summary (see Deleting a Specification from the Spec Summary on page 755).

Recreating the Spec Summary from the Results in the Active Results Tab

To clear the current spec summary view and display all the specifications from the active Results tab, do the following.

➤ Click the  button.

Note: You can recreate the spec summary only from the results of a Single Run, Sweeps and Corners or Monte Carlo simulation run. You cannot recreate the spec summary from the results of a Global Optimization, Local Optimization, , Feasibility Analysis, Size Over Corners or Sensitivity Analysis simulation run.

Viewing the Detailed Results for Specifications

To view the detailed results for a specification, do the following:

➤ Select a specification and click the  button.

To view the detailed results for multiple specifications, hold down the *Shift* key (for contiguous selection) or the *Ctrl* key (for noncontiguous selection) and click the  button. Ensure that the specifications you select have the same conditions.

The detailed results are displayed in the detail view. You can click the  button to view the detailed results in horizontal or vertical format.

Figure 15-5 Detailed Results in Horizontal Format

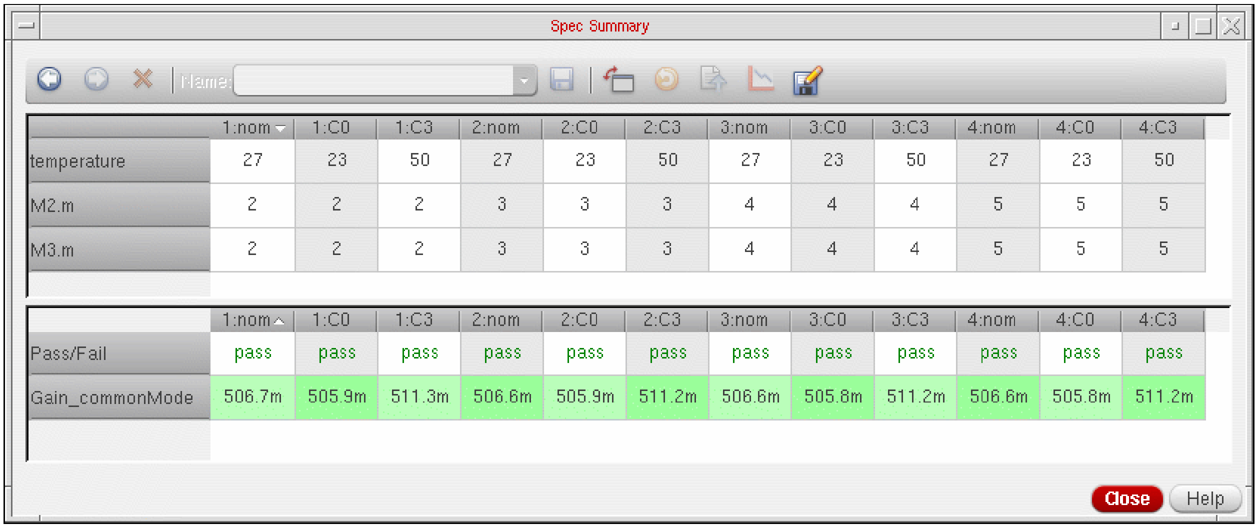
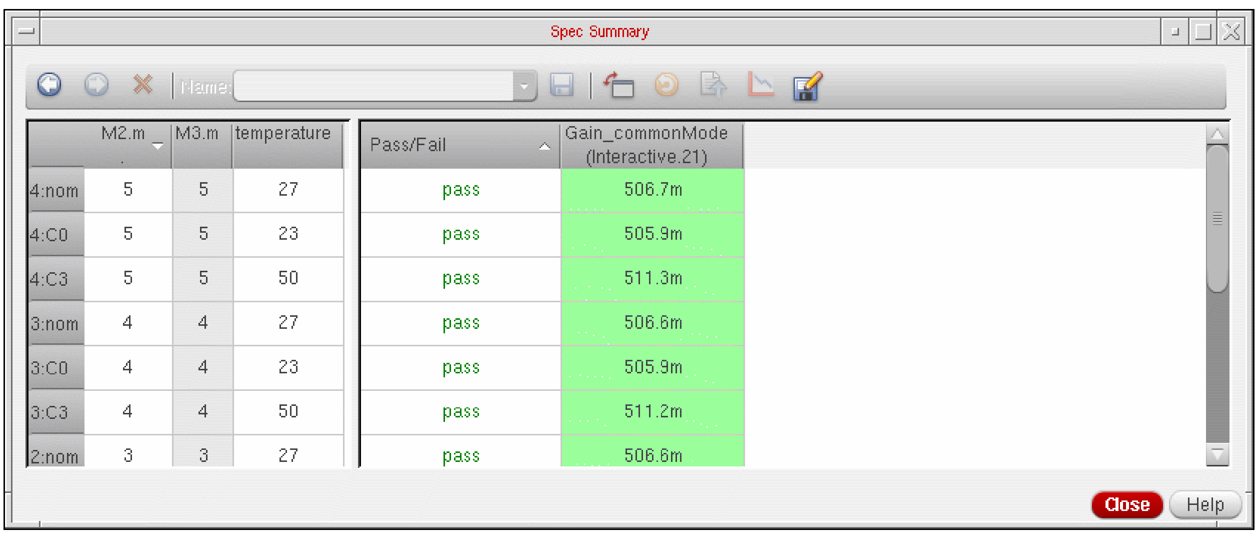


Figure 15-6 Detailed Results in Vertical Format



The detail view displays the following information:

- ❑ Values of swept parameters for each simulation.
- ❑ *pass*, *near or fail* status for each simulation.
  - ❍ *pass* means that the measured value is within the limits defined by the specification.
  - ❍ *near* means that the measured value is no more than 10% outside the target value of the specification.

  - ❍ *fail* means that the measured value is greater than 10% outside the target value of the specification.

- ❑ Measured values from each simulation.

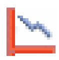
  The measured values with a *pass* status are displayed with a green background, those with a *near* status appear with a yellow background, and those with a *fail* status appear with a red background.

To go back to the summary view, do the following:

➤ Click the  button.

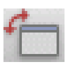
Plotting the Results for Specifications

To plot the results for the specifications in the summary view, do the following:

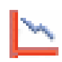
➤ Select a specification and click the  button.

To plot the results for multiple specifications, hold down the *Shift* key (for contiguous selection) or the *Ctrl* key (for noncontiguous selection) and click the  button.

To plot the results in the detail view, do the following:

1. Click the  button to display the results in the detail view in the vertical format.

   See Detailed Results in Vertical Format figure on page 758 for an example of the detail in the vertical format.

2. Select a row in the table on the left-hand side and click the  button.

   To plot the results for multiple rows, hold down the *Shift* key (for contiguous selection) or the *Ctrl* key (for noncontiguous selection) and click the  button.

Exporting a Spec Summary to a HTML or CSV File

To export a spec summary to a HTML or comma-separated values (CSV) file, do the following:

1. Click the  button.

   The Export Results form appears.

2. In the *File name* field, type a file name with the `.html`, `.htm` or `.csv` extension.

3. Click *Save*.

The program exports the results to the file you specified. By default, the file is saved in the `documents` folder of the ADE XL view and displayed in the *Documents* tree on the Data View pane. For more information about working with documents, see Chapter 18, "Working with Documents."

Note: If you click the  button in the detail view, the resulting HTML or CSV file contains the contents of the detail view and summary view.

Sorting Data in the Spec Summary Form

To sort data in the Spec Summary form, do the following:

➤ Click on the name of the column based on which you want to sort the data.

Hiding and Showing Columns in the Spec Summary Form

To hide a column, do the following:

➤ Right-click on the name of any column and select the name of the column you want to hide.

To display a hidden column, do the following:

➤ Right-click on the name of any column and select the name of the column you want to display.

16

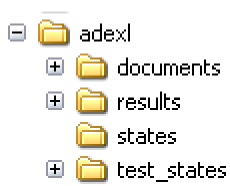
Working with the Simulation Setup

This chapter describes the following topics:

ADE XL View Directory Structure

The simulation setup in ADE XL is saved in the ADE XL view. The default name for the ADE XL view is `adexl`. The ADE XL view has the following directory structure:

Figure 16-1 ADE XL View Directory Structure



The following table describes the contents of folders in the ADE XL view.

Table 16-1 Contents of ADE XL View

Folder	Description
documents	Contains the documents you added in ADE XL and the specification summaries and datasheets you created in ADE XL. For information about working with documents, see Chapter 18, “Working with Documents.” For information about creating a specification summary, see Working with the Specification Summary on page 749. For information about creating datasheets, see Working with Datasheets on page 820.

Table 16-1 Contents of ADE XL View

Folder	Description
results	Contains the results database (`.rdb`) files and run log (`.log`) files for history items. For example, the `Interactive.1.rdb` and `Interactive.1.log` files in the `results` folder are the results database and run log files for the history item `Interactive.1`. For more information about working with history items, see Chapter 17, "Working with Checkpoints." To specify a different location where you want the program to save the results database and run log files, do one of the following: ■ Use the *ADE XL Results Database Location* field in the Save Options form. For more information, see Specifying Results Database Location on page 76 ■ Use the `adexl.results saveDir` environment variable. For more information, see saveDir on page 899. **Note:** If you do not specify a different location, and you open the ADE XL view in read-only mode or you do not have write permissions in the ADE XL view, the program writes this information to *`libraryName`*`/`*`cellName`*`/adexl/results/data` in the location specified using the `asimenv.startup projectDir` environment variable. The default setting for this environment variable is `$HOME/simulation`. **Note:** Simulation results are not saved in the `results` folder. For information about the location of simulation results information, see Specifying Options for Saving Simulation Results on page 73.
states	Contains the setup states you created in ADE XL. For more information, see Working with Setup States on page 783.

Table 16-1 Contents of ADE XL View

Folder	Description
test_states	Contains the ADE states for the current simulation setup and the simulation setup for history items. **Note:** If you open the ADE XL view in read-only mode or you do not have write permissions in the ADE XL view, the program writes this information to `*libraryName*/*cellName*/adexl/results/data/<history_item>` in the location specified using the `asimenv.startup projectDir` environment variable. The default setting for this environment variable is `$HOME/simulation`.

Saving the Simulation Setup

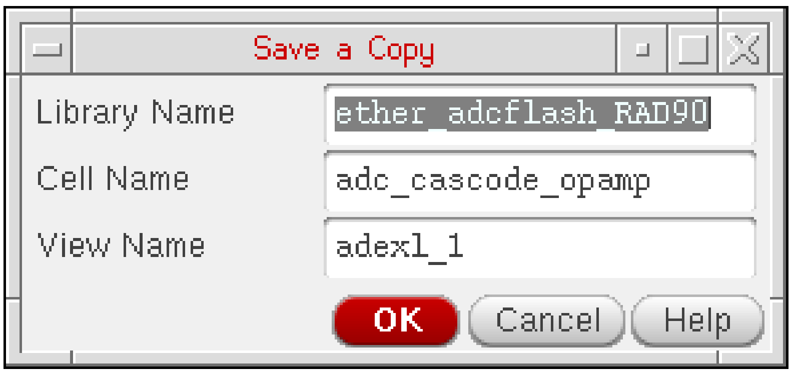
To save the simulation setup, do the following

➤ Choose *File – Save*.

To save the simulation setup to a different ADE XL view, do the following:

1. Choose *File – Save As*.

   The Save As form appears.



2. Enter the name of the library, cell and view in which you want to save the setup.
3. Click *OK*.

Note: This action does not save history data, only the setup database.

See also

- Saving Setup Changes in Read-Only ADE XL Views on page 778

Copying an ADE XL View

Caution

Do not copy ADE XL views using file system commands or a graphical File Manager. This can result in loss of data from previous simulation runs.

You can use Library Manager to copy ADE XL views. Use one of the methods described in Table 16-1 when you:

- Copy an ADE XL view
- Copy a cell containing an ADE XL view
- Copy a library in which any cell has an ADE XL view

Note: This section describes only the procedures for copying an ADE XL view. The procedures for copying libraries or cells containing ADE XL views are similar. For more information about copying libraries and cells, see the *Cadence Library Manager User Guide*.

Table 16-2 Methods of Copying ADE XL Views

Method	Purpose
Copy only the simulation setup, but not the results database, run log files, and simulation results for the history items.	Use this method if you only want to reuse the simulation setup existing in an ADE XL view, and do not want to view the simulation results, plot waveforms, or do other tasks that need access to simulation results in the new ADE XL view. For more information, see Copying only the Simulation Setup in an ADE XL View on page 767. **Note:** If the results database exists within the ADE XL view that you are copying, you can view the simulation results in the new ADE XL view. However, you cannot plot waveforms or do other tasks that need access to simulation results in the new ADE XL view. For more information about the location of the results database, see Specifying Results Database Location on page 76.

Method	Purpose
Copy the simulation setup and the results database	Use this method if you want to reuse the simulation setup and view the simulation results, but do not want to plot waveforms or do other tasks that need access to simulation results in the new ADE XL view. For more information, see Copying the Simulation Setup and the Results Database in an ADE XL View on page 770.
Copy everything (the simulation setup, results database, run log files and the simulation results) in the ADE XL view	Use this method if you want to make a complete copy of an ADE XL view, or create a backup of an ADE XL view. For more information, see Copying Everything in an ADE XL View on page 772.

Copying only the Simulation Setup in an ADE XL View

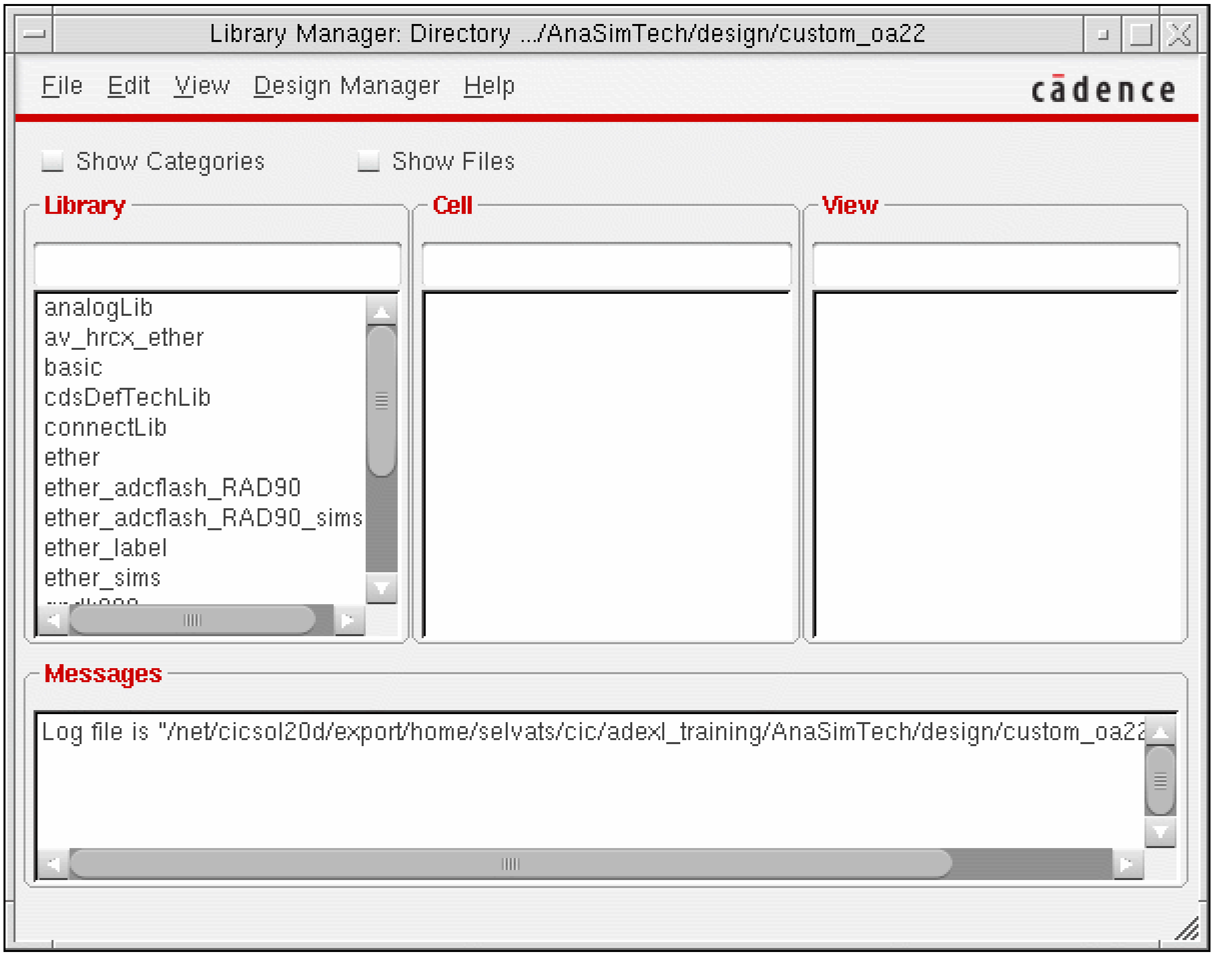
To copy only the Simulation Setup in an ADE XL View, do the following.

1. Set the following environment variables in your `.cdsenv` file:

```
ddserv.lib enableCopyInDFII boolean t
cdsLibManager.copyGlobals mpsRadio toggle t
```

2. Start the Virtuoso design environment.
3. In the CIW, choose *Tools – Library Manager*.

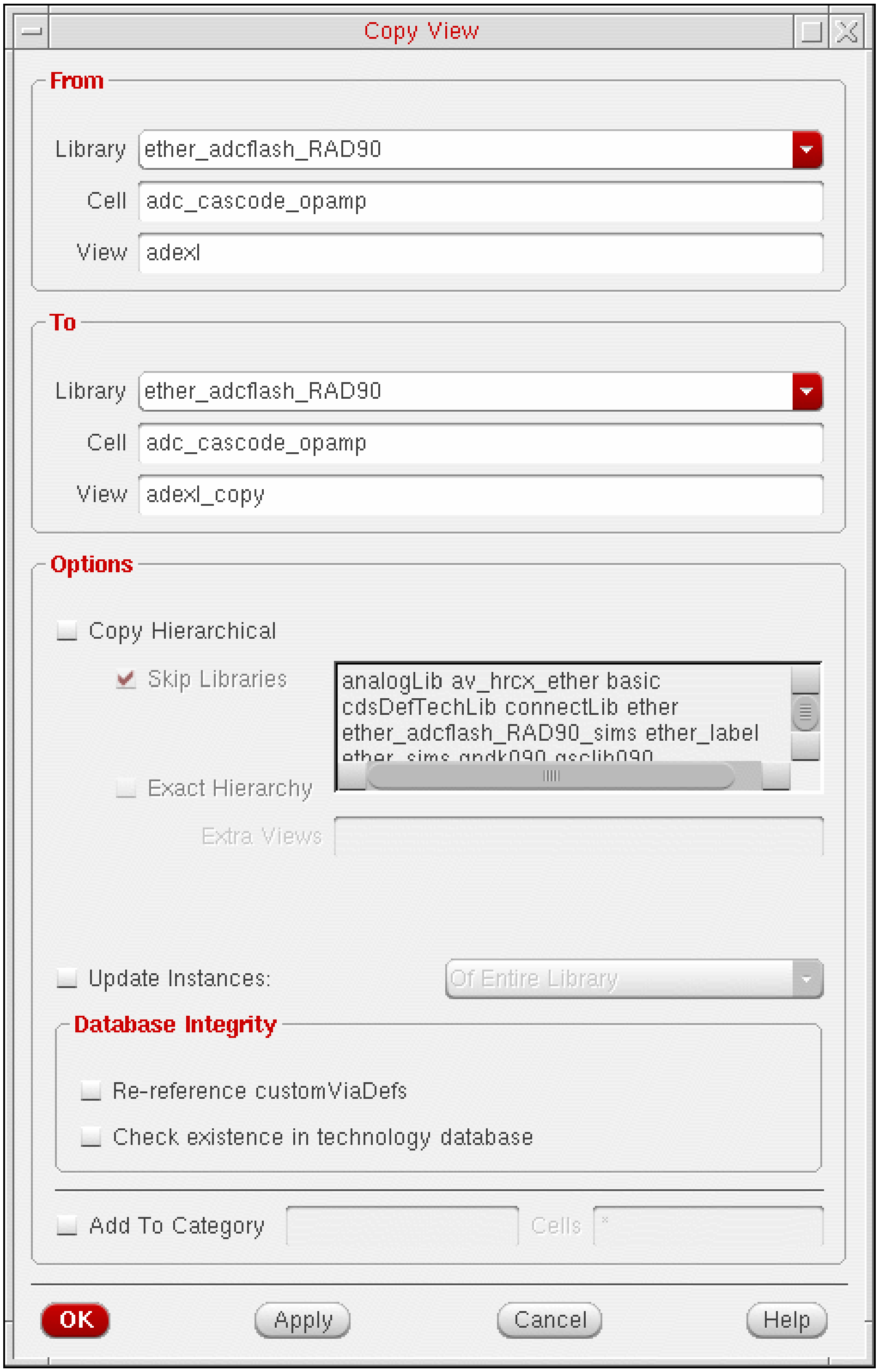
The Library Manager form appears.



4. In the *Library* column, select the library in which the ADE XL view exists.
5. In the *Cell* column, select the cell in which the ADE XL view exists.
6. In the *View* column, select the ADE XL view.
7. Choose *Edit – Copy*.

   The Copy View form appears.

8. In the *To* group box, type specify the destination library, cell and view name.



9. Click *OK* to copy the ADE XL view.

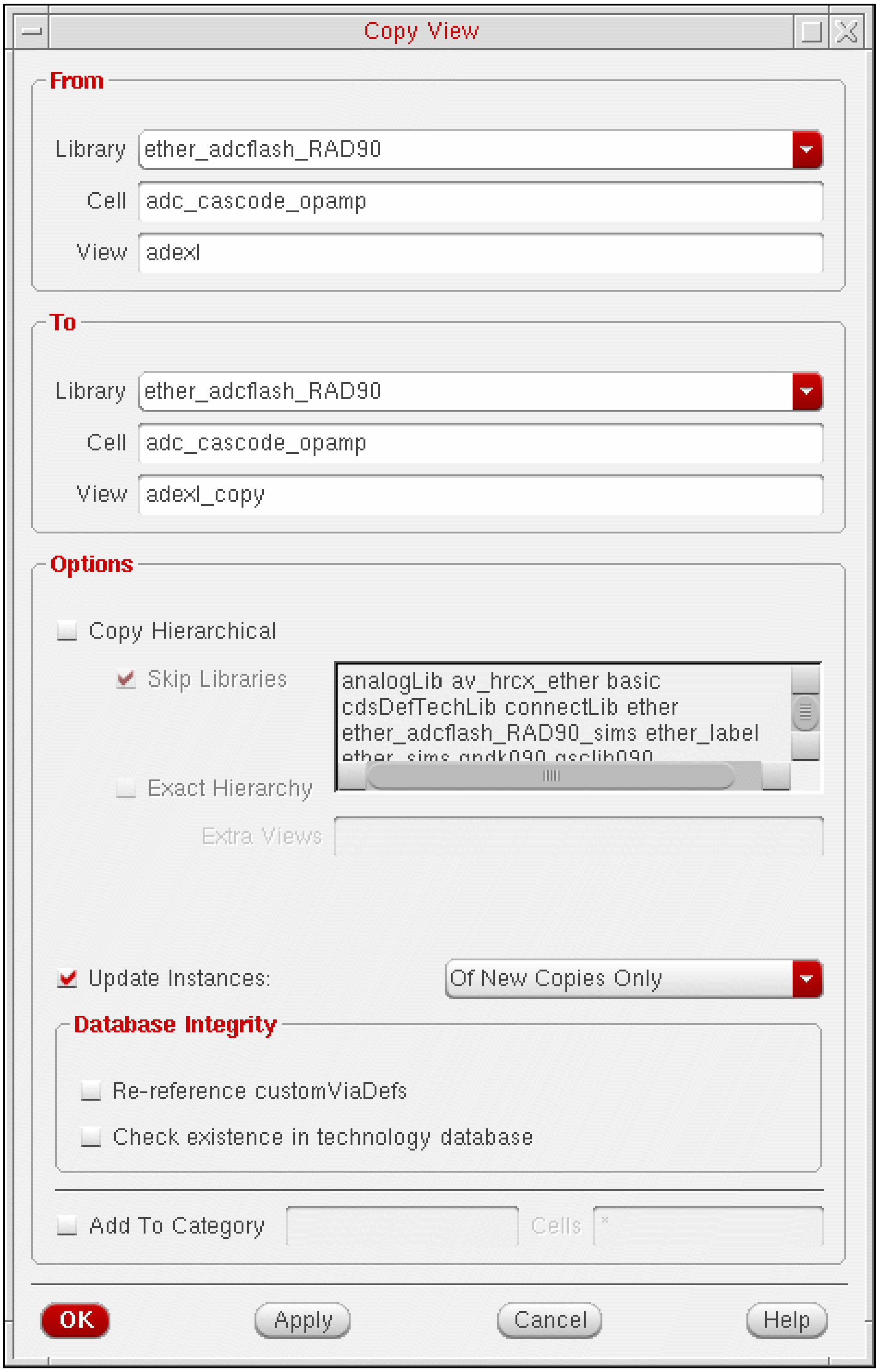
Copying the Simulation Setup and the Results Database in an ADE XL View

To copy the simulation setup and the results database in an ADE XL view, do the following:

1. Perform steps 3 to 8 described in Copying only the Simulation Setup in an ADE XL View on page 767.
2. Select the *Update Instances* check box.

   The drop-down list becomes active.

3. From the drop-down list, select *Of New Copies Only*.



4. Click *OK* to copy the ADE XL view.

See also:

■ How is the Results Database is Copied When an ADE XL View is Copied? on page 772

Copying Everything in an ADE XL View

To copy everything in an ADE XL view, do the following:

1. If you have used the `asimenv.startup projectDir` environment variable to specify the location for storing simulation results, ensure that the variable is set in your `.cdsenv` file.

   For more information about specifying the location for storing simulation results, see the following topics:

   ❑ Specifying Options for Saving Simulation Results on page 73

   ❑ saveResDir on page 901

2. Set the `adexl.cpupdtr copyResultsData` environment variable in your `.cdsenv` file.

3. Start the Virtuoso design environment.

4. Copy the ADE XL view using the procedure described in Copying the Simulation Setup and the Results Database in an ADE XL View on page 770.

See also:

■ How is the Results Database is Copied When an ADE XL View is Copied? on page 772

■ How are Simulation Results Copied When an ADE XL View is Copied? on page 773

How is the Results Database is Copied When an ADE XL View is Copied?

This section describes how the results database is copied when an ADE XL view is copied using one of the following procedures:

■ Copying the Simulation Setup and the Results Database in an ADE XL View on page 770

■ Copying Everything in an ADE XL View on page 772

The results database can be located inside or outside the ADE XL view. For more information about the location of the results database, see Specifying Results Database Location on page 76.

- If the results database is located inside the ADE XL view, it is copied into the new ADE XL view.
- If the results database is located outside the ADE XL view, it will not be copied into the new ADE XL view. Instead, a copy of the results database will be created in the specified results database location. For example, if the results database for the `adexl` view in the `myCell` cell of the `myLib` library is located outside the `adexl` view at:

  `$HOME/myResultsDatabase/myLib/myCell/adexl/results/data`

  and you copy the `adexl` view to a new view called `adexl_copy`, the results database for the `adexl_copy` view will be located at:

  `$HOME/myResultsDatabase/myLib/myCell/adexl_copy/results/data`

How are Simulation Results Copied When an ADE XL View is Copied?

Simulation results are always located outside the ADE XL view. For more information about the location of simulation results, see Specifying Options for Saving Simulation Results on page 73.

When an ADE XL view is copied using the procedure described in Copying Everything in an ADE XL View on page 772, simulation results will not be copied into the new ADE XL view. Instead, a copy of the simulation results is created in the specified simulation results location. For example, if the simulation results for the `adexl` view in the `myCell` cell of the `myLib` library is located outside the `adexl` view at:

`$HOME/mySimulationResults/myLib/myCell/adexl/results/data/<history_item>`

and you copy the `adexl` view to a new view called `adexl_copy`, the simulation results for the `adexl_copy` view will be located at:

`$HOME/myResultsDatabase/myLib/myCell/adexl_copy/results/data/<history_item>`

Deleting an ADE XL View

To delete an ADE XL view, do the following:

1. In the CIW, choose *Tools – Library Manager*.

   The Library Manager form appears.

2. In the *Library* column, select the library in which the ADE XL view exists.
3. In the *Cell* column, select the cell in which the ADE XL view exists.
4. In the *View* column, select the ADE XL view.
5. Choose *Edit – Delete*.

   The Delete Cell Views form appears.

6. Click *OK*.

Working with Read-Only ADE XL Views

An ADE XL view is considered a read-only view if:

- You open the ADE XL view in read-only mode.
- You do not have write permissions to the ADE XL view.

The following sections describe how you can work with read-only ADE XL views:

Opening ADE XL Views in Read-Only Mode

You can open an ADE XL view in read-only mode from the Open ADE (G)XL View form or from Library Manager.

Note: You need an ADE XL and VSE L/XL license to open a ADE XL view in read only mode. (CCR 3561651)

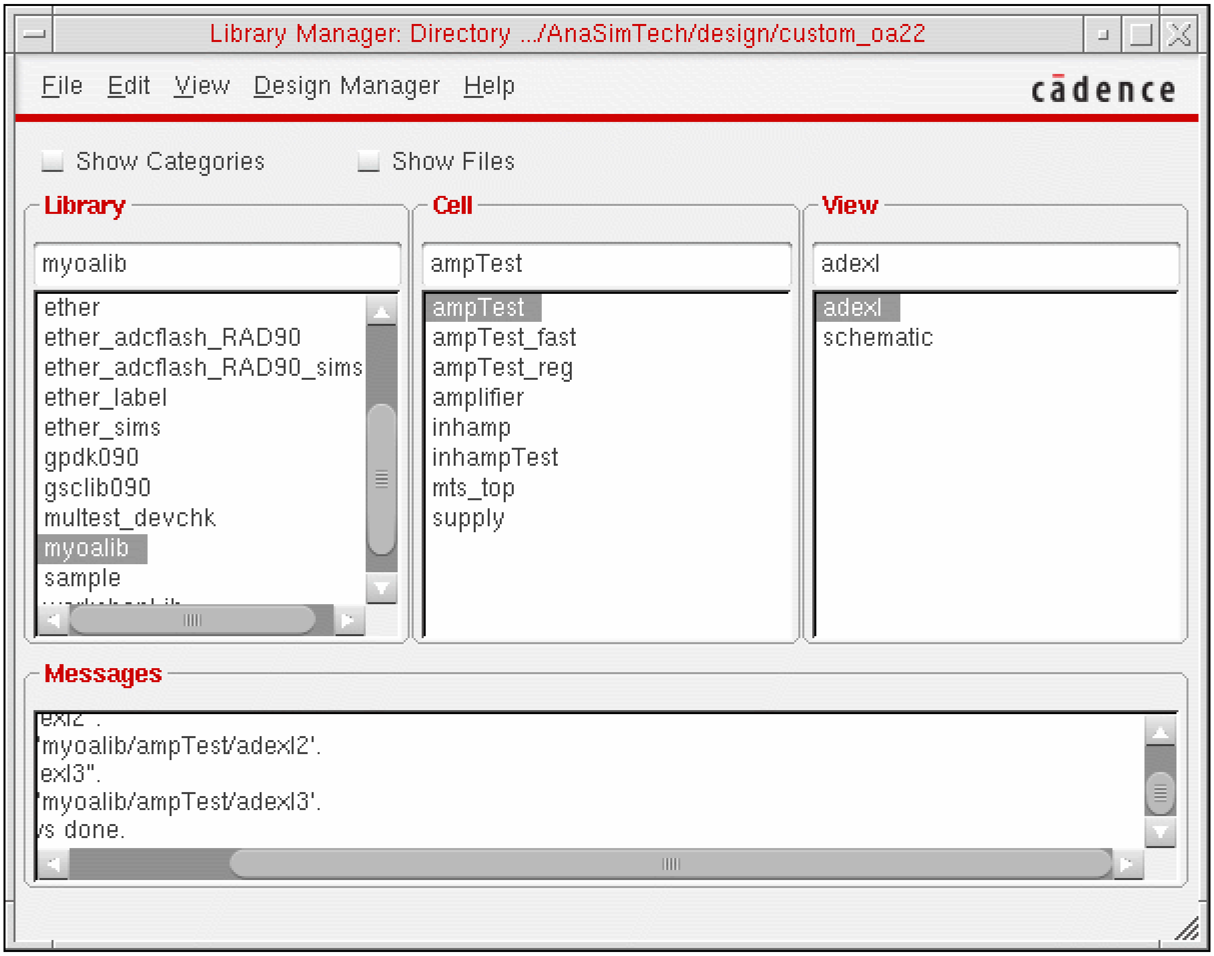
For information about opening an ADE XL view in read-only mode from the Open ADE (G)XL View form, see Opening an Existing Setup on page 39.

To open an ADE XL view in read-only mode from Library Manager, do the following:

1. In the CIW, choose *Tools – Library Manager*.

   The Library Manager form appears.

2. Use the *Library*, *Cell*, and *View* fields to select your ADE XL view.



3. In the *View* field, right-click on the ADE XL view and choose *Open (Read -Only)*.

   The view is opened in ADE XL.

The ADE XL title bar displays the text *Reading* if the view is opened in read-only mode.

Opening ADE XL Views to Which You Do Not Have Write Permissions

When you open an ADE XL view to which you do not have write permissions, the following message appears:

```
Could not open <library_name> <cell_name> <view_name> for edit (no data or lock
available). Would you like to open it for read?
```

Click *Yes* to open the ADE XL view in read-only mode.

For more information about opening ADE XL views, see “Opening an Existing Setup” on page 39.

Running Simulations from Read-Only ADE XL Views

You can run simulations from a read-only ADE XL view. However, note the following when you run simulations from read-only ADE XL views:

- Location of Simulation Results Information for Read-Only ADE XL Views on page 777
- Location of Results Database Information for Read-Only ADE XL Views on page 777
- Number of History Entries Saved for Read-Only ADE XL Views on page 778

Note: You can disable running of simulations in a read-only ADE XL view using the `adexl.gui disableRunInReadOnly` environment variable.

Location of Simulation Results Information for Read-Only ADE XL Views

By default, the program writes simulation results information to `*libraryName*/*cellName*/adexl/results/data/<*history_item*>` in the location specified using the `asimenv.startup projectDir` environment variable. The default setting for this environment variable is `$HOME/simulation`.

You can specify a different location for storing simulation results using the procedure described in Specifying Options for Saving Simulation Results on page 73.

Location of Results Database Information for Read-Only ADE XL Views

By default, the program writes results database information to `*libraryName*/*cellName*/adexl/results/data` in the location specified using the `asimenv.startup projectDir` environment variable. The default setting for this environment variable is `$HOME/simulation`.

You can specify a different location for storing results database information using the procedure described in Specifying Results Database Location on page 76.

Number of History Entries Saved for Read-Only ADE XL Views

The setting for the number of history entries to be saved, specified in the Save Options form or the `saveLastNHistoryEntries` environment variable, will only apply to history entries created in the current ADE XL session. History entries created in previous ADE XL sessions will not be automatically removed.

For more information about using the Save Options form, see Specifying How Much Data to Save on page 805. For more information about the `saveLastNHistoryEntries` environment variable, see saveLastNHistoryEntries on page 859.

Saving Setup Changes in Read-Only ADE XL Views

If you make any changes to the setup in a read-only ADE XL view, the Save As form appears when you save or close the ADE XL view. You can save the changes to a new ADE XL view in a location where you have write permissions.

For information about using the Save As form, see Saving the Simulation Setup on page 765.

Importing and Exporting the Simulation Setup

You can import the simulation setup from an ADE XL view into the current ADE XL view. You can also export the simulation setup from the current ADE XL view to an existing or new ADE XL view. You can import or export the complete simulation setup or only specific simulation settings. This allows you to quickly reuse the simulation setup created for other designs.

For more information, see the following sections:

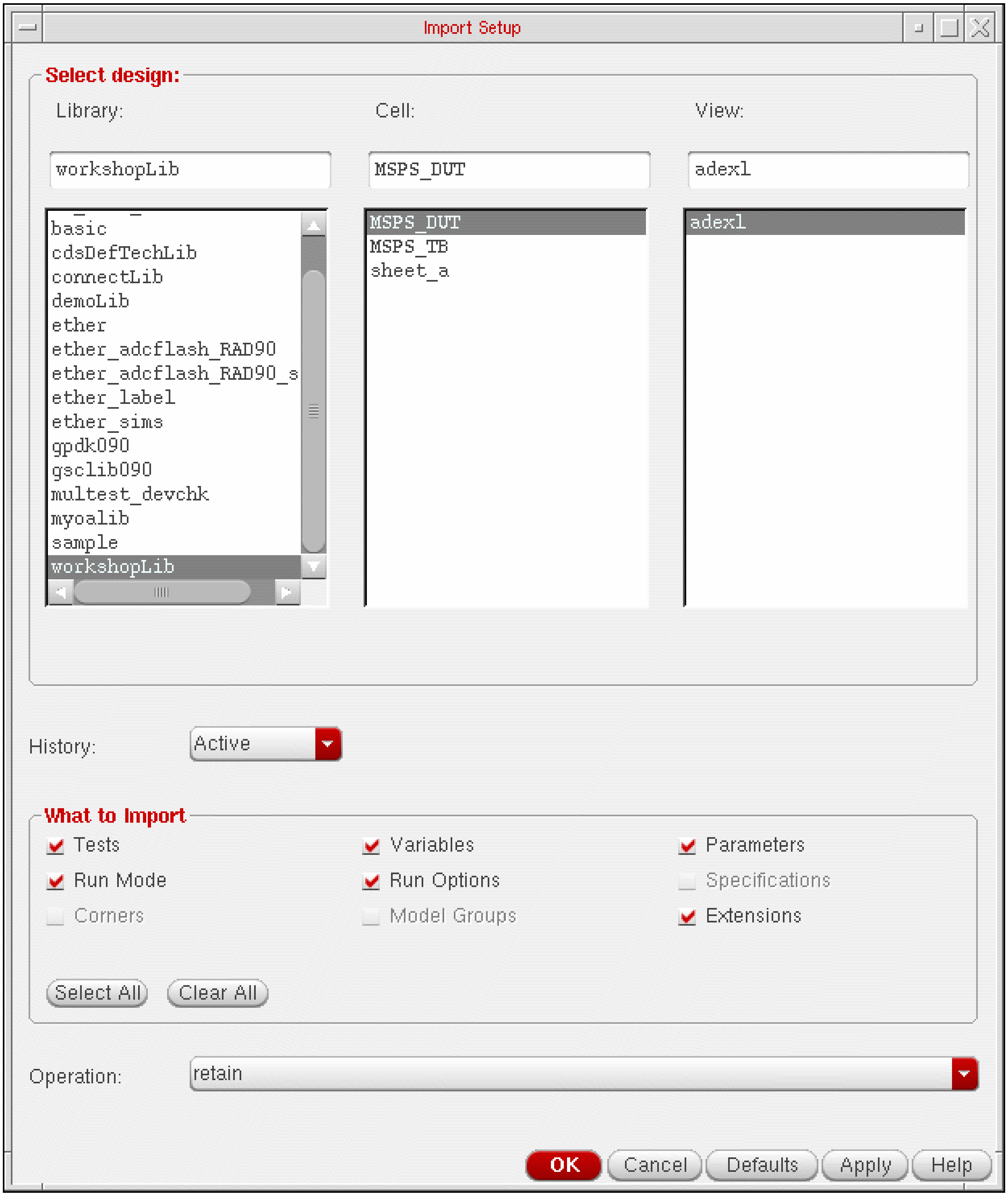
- Importing the Simulation Setup on page 778
- Exporting the Simulation Setup on page 781

Importing the Simulation Setup

To import the simulation setup from an existing ADE XL view to the current ADE XL view, do the following:

1. Choose *File – Import*.

The Import Setup form appears.



2. Select the ADE XL view from which you want to import the setup.

3. In the *History* drop-down list, select the history item from which you want to import the setup. By default, the current simulation setup (indicated by the text *Active*) is selected for import.
4. In the *What to Import* group box, select the check box next to the settings you want to import.

   By default, all the settings existing in the history item specified in the *History* drop-down list are selected for import. You can click *Select All* to select all the settings for import, or click *Clear All* to clear all the selections.

Select	To
Tests	Import the test definitions in the Data View pane. The simulation state of each test is also saved in the subdirectories.
Variables	Import the global variable definitions, including their enabled status and values, in the Data View pane.
Parameters	Import the deifinitions of parameters, including their enabled status and values, in the Data View pane.
Run Mode	Import the active run mode and set it on the Run toolbar.
Run Options	Import the following settings: ■ Options in the Run Options form. ■ Simulation options for all the customized run modes. For example, if you changed the default simulation options of the Monte Carlo, Global Optimization, and Local Optimization run modes before saving the state, the tool will import the settings for all the three run modes.
Specifications	Import the specifications defined in the Outputs pane. This includes the details of the specification type and their target values.
Corners	Import the corner definitions defined in the Corners Setup form.
Model Groups	Import the model groups defined in the Corners Setup form.
Extensions	Import the parasitic details in the Parasitic Setup form.

5. In the *Operation* drop-down list, select the import mode.

Select	To
retain	Retain the current setup information and append other setup information for each setting selected in the *What to Import* group box. For more information, see How the Simulation Setup is Updated When You Load Setup States on page 791.
merge	Overwrite the current setup information for items that have the same name and append other setup information for each setting selected in the *What to Import* group box. For more information, see How the Simulation Setup is Updated When You Load Setup States on page 791.
overwrite	Overwrite the current setup information for each setting selected in the *What to Import* group box. For more information, see How the Simulation Setup is Updated When You Load Setup States on page 791.

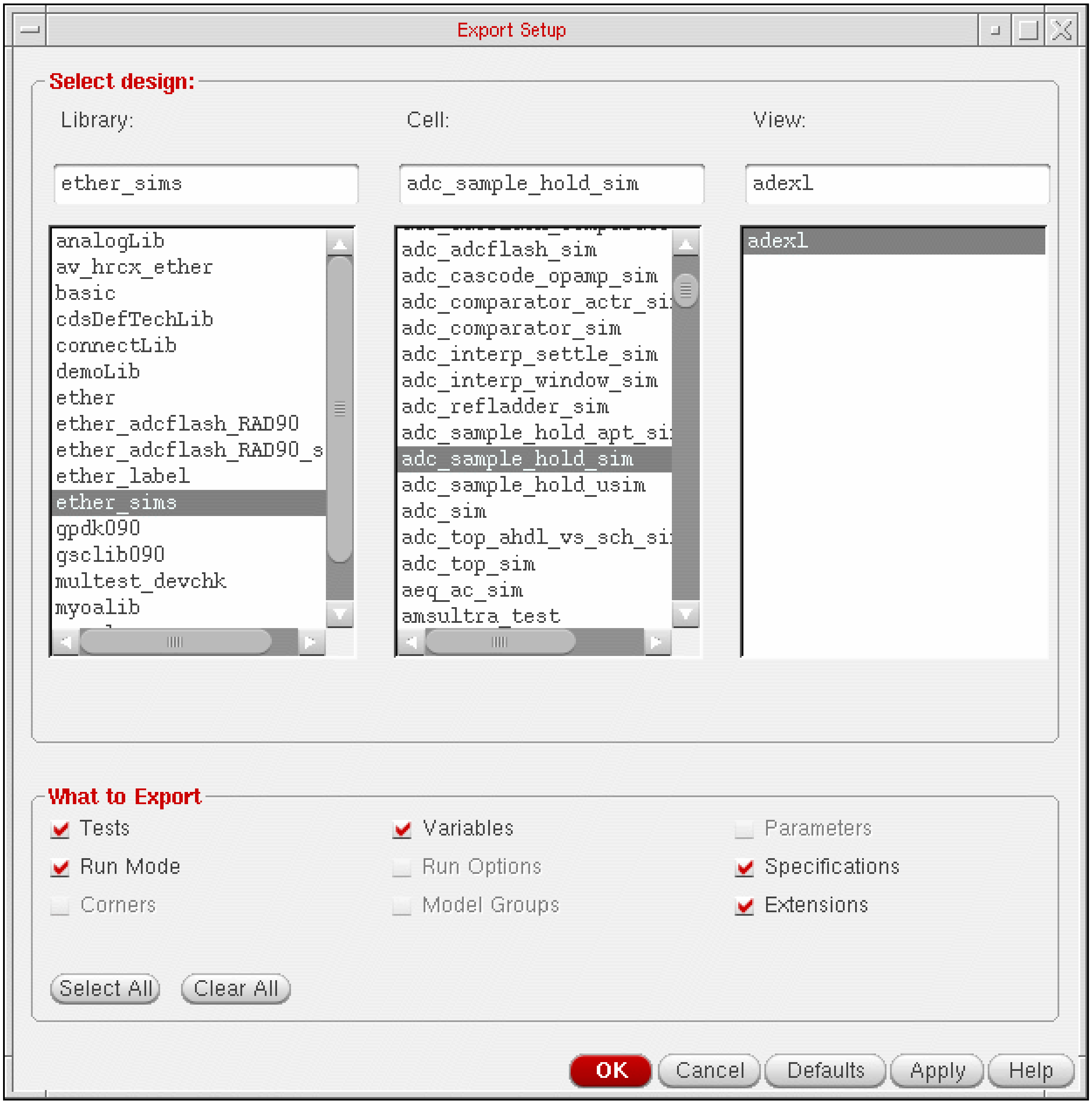
6. Click *OK* to import the setup into the current ADE XL view.

Exporting the Simulation Setup

To export the simulation setup from the current ADE XL view to an existing ADE XL view, do the following:

1. Choose *File – Export*.

The Export Setup form appears.



2. Select the ADE XL view to which you want to export the setup.
3. In the *What to Export* group box, select the check box next to the settings you want to export.

By default, all the settings existing in the current ADE XL view are selected for export. You can click *Select All* to select all the settings for import, or click *Clear All* to clear all the selections.

Select	To
Tests	Export the test definitions in the Data View pane. The simulation state of each test is also saved in the subdirectories.
Variables	Export the global variable definitions, including their enabled status and values, in the Data View pane.
Parameters	Export the deifinitions of parameters, including their enabled status and values, in the Data View pane.
Run Mode	Export the active run mode set on the Run toolbar.
Run Options	Export the following settings: ■ Options from the Run Options form. ■ Simulation options for all the customized run modes. For example, if you have changed the default settings of the Monte Carlo and Local Optimization run modes, but the currently active run mode is Global Optimization, the tool will export the settings for all three run modes.
Specifications	Export the specifications defined in the Outputs pane. This includes the details of the specification type and their target values.
Corners	Export the corner definitions defined in the Corners Setup form.
Model Groups	Export the model groups defined in the Corners Setup form.
Extensions	Export the parasitic details set by using the Parasitic Setup form.

4. Click *OK* to export the setup to the specified ADE XL view.

Working with Setup States

ADE XL and ADE GXL support different run modes like Single Run, Sweeps and Corners, Monte Carlo Sampling, Local Optimization, Global Optimization, and so on. For more information about run modes, see Specifying the Run Mode on page 71.

For each run mode, you may need a different simulation setup. For example, you may need different parameter specifications for running Monte Carlo analysis, optimization and . Similarly, you may need a different set of global variable values for running local and global optimization. Even for the same run mode, you may need a different simulation setup. For example, you may want to run Single Run, Sweeps and Corners with a different set of global variables.

ADE XL and ADE GXL allow you to create setup states that contain all or part of the simulation setup. You can later restore the simulation setup from the setup state by loading all or part of the settings in the setup state. This allows you to avoid modifying the simulation setup every time you need to run simulation using a different setup. The following examples describe the use of setup states:

Example 16-1 Different Setup States for Different Run Modes

To have a different setup for Single Run, Sweeps and Corners runs and Monte Carlo analysis, do the following:

1. Define the settings required for Single Run, Sweeps and Corners runs and save the settings in a setup state named `SweepsCornersSetup`.

   For more information about creating a setup state, see Creating or Updating a Setup State on page 785.

2. Modify the settings as required for Monte Carlo analysis, and save the settings in a setup state named `MonteCarloSetup`.

3. Do the following:

   ❑ Before you run Single Run, Sweeps and Corners, load the `SweepsCornersSetup` setup state using the *Overwrite* option.

   For more information about loading a setup state, see Loading a Setup State on page 787.

   ❑ Before you run Monte Carlo analysis, load the `MonteCarloSetup` setup state using the *Overwrite* option.

Example 16-2 Different Setup States for Different Set of Global Variables

To run Single Run, Sweeps and Corners with a different set of global variables, do the following:

1. Create the first set of global variables and save the variables in a setup state named `SetupGlobalVars_1`.

For more information about creating global variables, see Creating a Global Variable on page 163. For more information about creating a setup state, see Creating or Updating a Setup State on page 785.

2. Delete the first set of global variables.

3. Create the second set of global variables and save the variables in a setup state named `SetupGlobalVars_2`.

4. Do one of following before you run Single Run, Sweeps and Corners:

   - To run Single Run, Sweeps and Corners using the global variables in the setup state named `SetupGlobalVars_1`, load the `SetupGlobalVars_1` setup state using the *Overwrite* option.

     For more information about loading a setup state, see Loading a Setup State on page 787.

   - To run Single Run, Sweeps and Corners using the global variables in the setup state named `SetupGlobalVars_2`, load the `SetupGlobalVars_2` setup state using the *Overwrite* option.

See the following topics for more information:

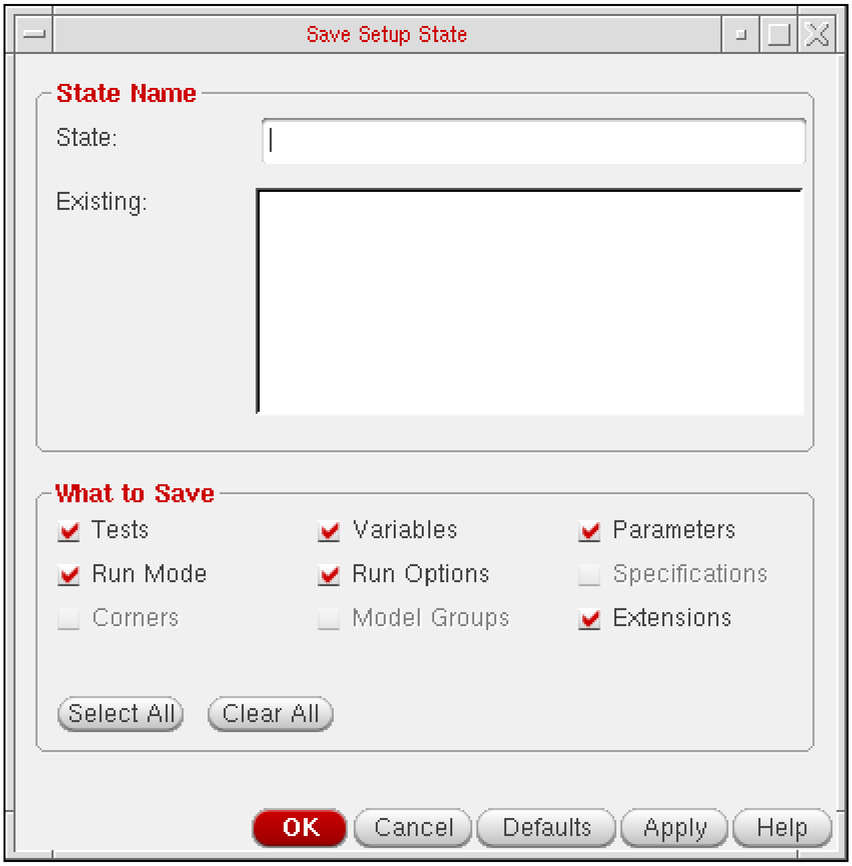
- Creating or Updating a Setup State on page 785
- Loading a Setup State on page 787
- Deleting a Setup State on page 790
- Running a Simulation Using a Setup State on page 795

Creating or Updating a Setup State

To create a new setup state, or update an existing setup state, do the following:

1. Choose *File – Save Setup State*.

The Save Setup State form appears.



2. Do one of the following:

   - To create a new setup state, enter the name of the setup state in the *State* field.
   - To update an existing setup state, select an existing state in the *Existing* list.

3. In the *What to Save* group box, select the check box next to the settings you want to save.

   By default, all the settings existing in the ADE XL view are selected for saving. You can click *Select All* to select all the settings for saving, or click *Clear All* to clear all the selections.

Select	To
Tests	Save the test definitions in the Data View pane. The simulation state of each test is also saved in the subdirectories.

Variables	Save the global variable definitions, including their enabled status and values, in the Data View pane.
Parameters	Save the deifinitions of parameters, including their enabled status and values, in the Data View pane.
Run Mode	Save the active run mode that is currently selected on the Run toolbar.
Run Options	Save the following settings: ■ Options from the Run Options form. ■ Simulation options for all the customized run modes. For example, if you have changed the default settings of the Monte Carlo and Local Optimization run modes, but the currently active run mode is Global Optimization, the tool will save the settings for all three run modes.
Specifications	Save the specifications defined in the Outputs pane. This includes the details of the specification type and their target values.
Corners	Save the corner definitions defined in the Corners Setup form.
Model Groups	Save the model groups defined in the Corners Setup form.
Extensions	Save the parasitic details set by using the Parasitic Setup form.

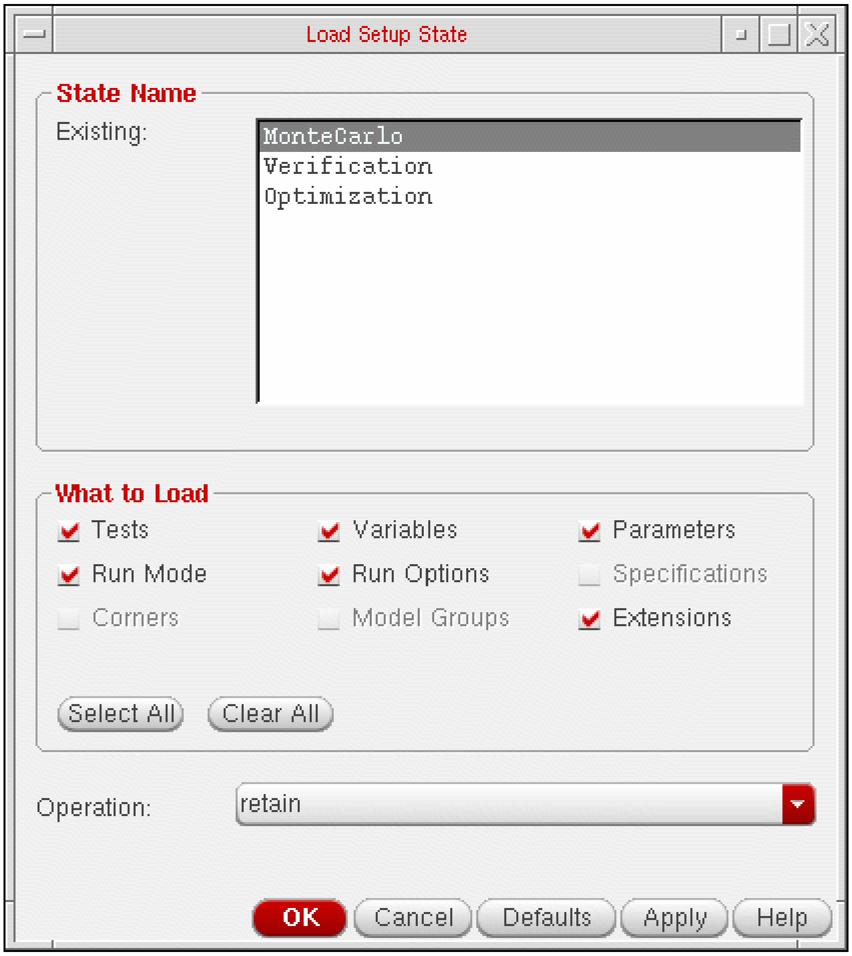
4. Click *OK*.

Loading a Setup State

To load a setup state, do the following:

1. Choose *File – Load Setup State*.

The Load Setup State form appears.



2. In the *State Name* group box, select the name of the state you want to load.
3. In the *What to Load* group box, select the check box next to the settings you want to restore to the current simulation setup.

   By default, all the settings existing in the setup state are selected for restore. You can click *Select All* to select all the settings for restoring, or click *Clear All* to clear all the selections.

Select	**To**
Tests	Restore the test definitions in the Data View pane. The simulation state of each test is also saved in the subdirectories.

Variables	Restore the global variable definitions, including their enabled status and values, in the Data View pane.
Parameters	Restore the deifinitions of parameters, including their enabled status and values, in the Data View pane.
Run Mode	Restore the active run mode and set it on the Run toolbar.
Run Options	Restore the following settings: ■ Options in the Run Options form. ■ Simulation options for all the customized run modes. For example, if you changed the default simulation options of the Monte Carlo, Global Optimization, and Local Optimization run modes before saving the state, the tool will restore the settings for all the three run modes.
Specifications	Restore the specifications defined in the Outputs pane. This includes the details of the specification type and their target values.
Corners	Restore the corner definitions and their enable status defined in the Corners Setup form.
Model Groups	Restore the model groups defined in the Corners Setup form.
Extensions	Restore the parasitic details set by using the Parasitic Setup form.

4. In the *Operation* drop-down list, select the restore mode.

Select	To
retain	Retain the current setup information and append other setup information for each setting selected in the *What to Load* group box. For more information, see How the Simulation Setup is Updated When You Load Setup States on page 791.
merge	Overwrite the current setup information for items that have the same name and append other setup information for each setting selected in the *What to Load* group box. For more information, see How the Simulation Setup is Updated When You Load Setup States on page 791.

overwrite Overwrite the current setup information for each setting selected in the *What to Load* group box.

For more information, see How the Simulation Setup is Updated When You Load Setup States on page 791.

5. Click *OK* to load the selected setup state.

Deleting a Setup State

To delete a setup state, do the following:

1. Choose *File – Remove Setup State*.

   The Remove Setup State form appears.

2. In the *State Name* group box, select the name of the setup state you want to delete.
3. Click *OK* to delete the setup state.

How the Simulation Setup is Updated When You Load Setup States

When you load a setup state, you can choose to load the following settings:

- Tests
- Run mode settings
- Corners
- Global variables
- Run options
- Model groups
- Parameters
- Specifications
- Extensions

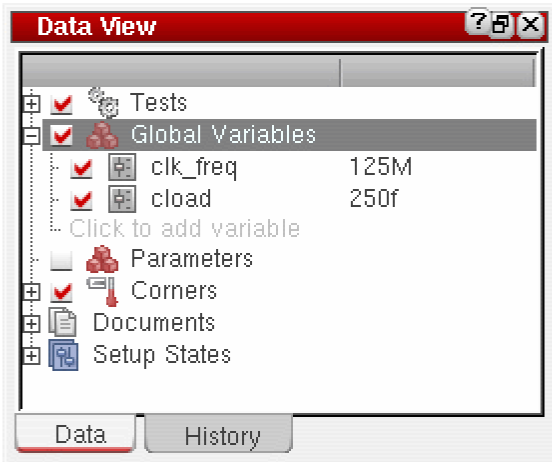
You can also choose to load setup states in the following modes:

- retain
- merge
- overwrite

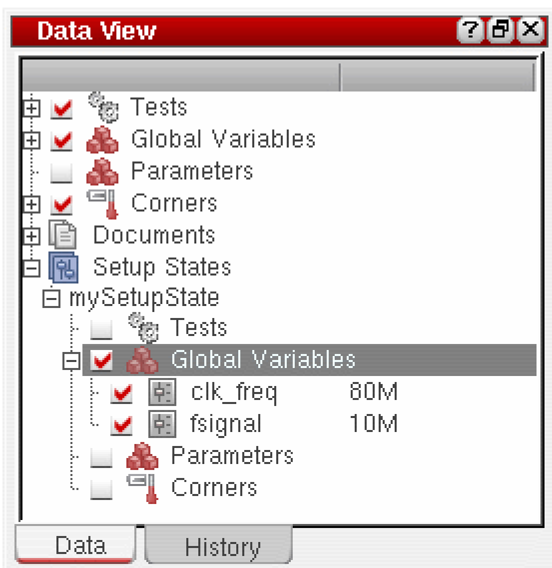
This section describes how the current simulation setup is updated when you load a setup state in `retain`, `merge` or `overwrite` mode. To understand how the current simulation setup is updated when you load a setup state in retain, merge or overwrite mode, we will do the following:

- Choose to load *Variables* from a setup state named `mySetupState`.

- Use a current simulation setup that has the following global variables:



- Load a setup state named `mySetupState` that has the following global variables:

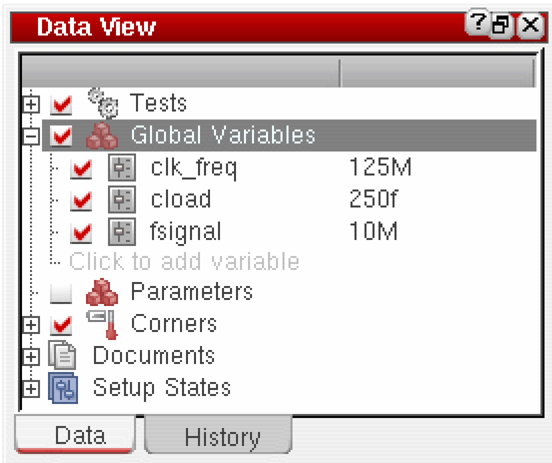


See the following sections to understand how the current simulation setup is updated when you load the setup state in retain, merge or overwrite mode:

- Results of Loading a Setup State in Retain Mode on page 793
- Results of Loading a Setup State in Merge Mode on page 794
- Results of Loading a Setup State in Overwrite Mode on page 795

Results of Loading a Setup State in Retain Mode

The updated simulation setup after loading the setup state in *retain* mode will have the following global variables:

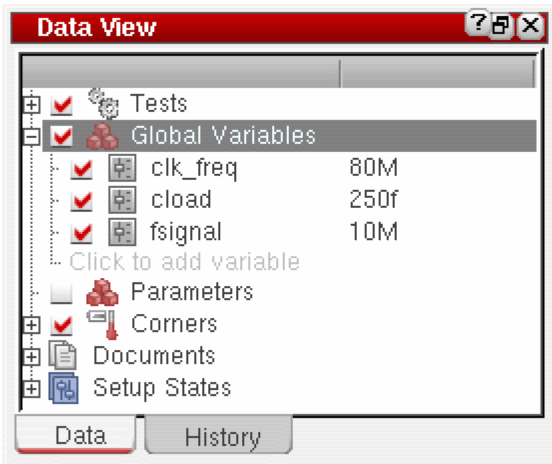


Note the following:

- The settings for the `clk_freq` variable are retained because the setup state has a variable with the same name.
- The settings for the `cload` variable are retained
- The `fsignal` variable is added because the current simulation setup does not have a variable with the same name

Results of Loading a Setup State in Merge Mode

The updated simulation setup after loading the setup state in *merge* mode will be:

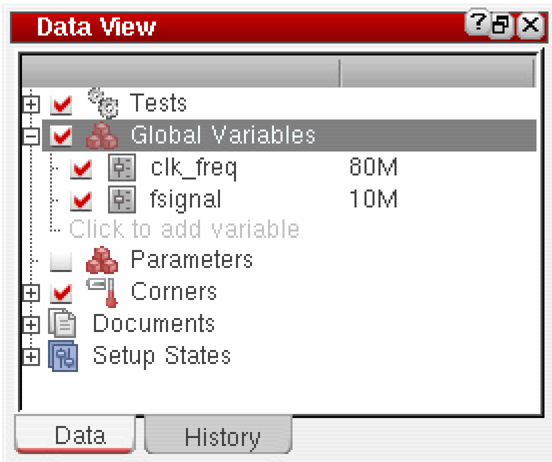


Note the following:

- The settings for the `clk_freq` variable are overwritten because the setup state has a variable with the same name.
- The settings for the `cload` variable are retained because the setup state does not have a variable with the same name.
- The `fsignal` variable is added because the current simulation setup does not have a variable with the same name.

Results of Loading a Setup State in Overwrite Mode

The updated simulation setup after loading the setup state in *overwrite* mode will be:



Note that all the global variables in the current simulation setup are deleted and the global variables in the setup state are added when you load the setup state in *overwrite* mode.

Running a Simulation Using a Setup State

You can run a simulation based on the settings in a setup state.

To run a simulation using a setup state, do the following:

➤ Right-click the setup state in the expanded *Setup States* tree on the Data View pane and choose *Run*.

Note: The history item for the simulation run will have a name starting with the name of the setup state. For example, if the name of the setup state is `myCorners`, the history item will have the name `myCorners.`*`seqNum`*, where *`seqNum`* is `0` (zero) for the first history item, then 1+(the largest existing *`seqNum`* for that setup state). For more information about history items, see Chapter 17, “Working with Checkpoints.”

Creating a Plan Using Setup States

A plan is a sequence of steps required to complete a particular task. For example, characterizing a design might involve running several tests or sweeps. These tasks can be grouped together to form a characterization plan.

You can create a plan by creating a setup state for each task in the plan and linking the setup states, in the order they need to be run, to an OCEAN script file. When you run the OCEAN script file, ADE XL does the following for each setup state, in the order in which the setup states are linked to the OCEAN script file:

1. Loads the current simulation setup to memory.
2. Loads the setup state to update the simulation setup in memory.
3. Runs simulation using the updated simulation setup in memory.
4. Clears the simulation setup in memory.

For example, if you have two tasks in a plan, create two setup states—say, `myTask1` and `mytask2`—that contain the simulation settings required to run the simulations corresponding to each task in the plan. Then link the setup states to an OCEAN script file named `myPlan.ocn`, in the order in which they need to be run. When you run the `myPlan.ocn` OCEAN script file, ADE XL does the following:

1. Loads the current simulation setup to memory.
2. Loads the `myTask1` setup state to update the simulation setup in memory.
3. Runs simulations using the updated simulation setup in memory.
4. Clears the simulation setup in memory.
5. Loads the current simulation setup to memory.
6. Loads the `myTask2` setup state to update the simulation setup in memory.
7. Runs simulations using the updated simulation setup in memory.
8. Clears the simulation setup in memory.

To create a plan, do the following:

1. Create setup states that contain the simulation settings required to run the simulations corresponding to each task in the plan. For more information, see Creating or Updating a Setup State on page 785.

2. Right-click a setup state in the expanded *Setup States* tree on the Data View pane and choose *Link to OCEAN XL Script*.

   The Choose an OCEAN XL File form appears.

3. Do one of the following:

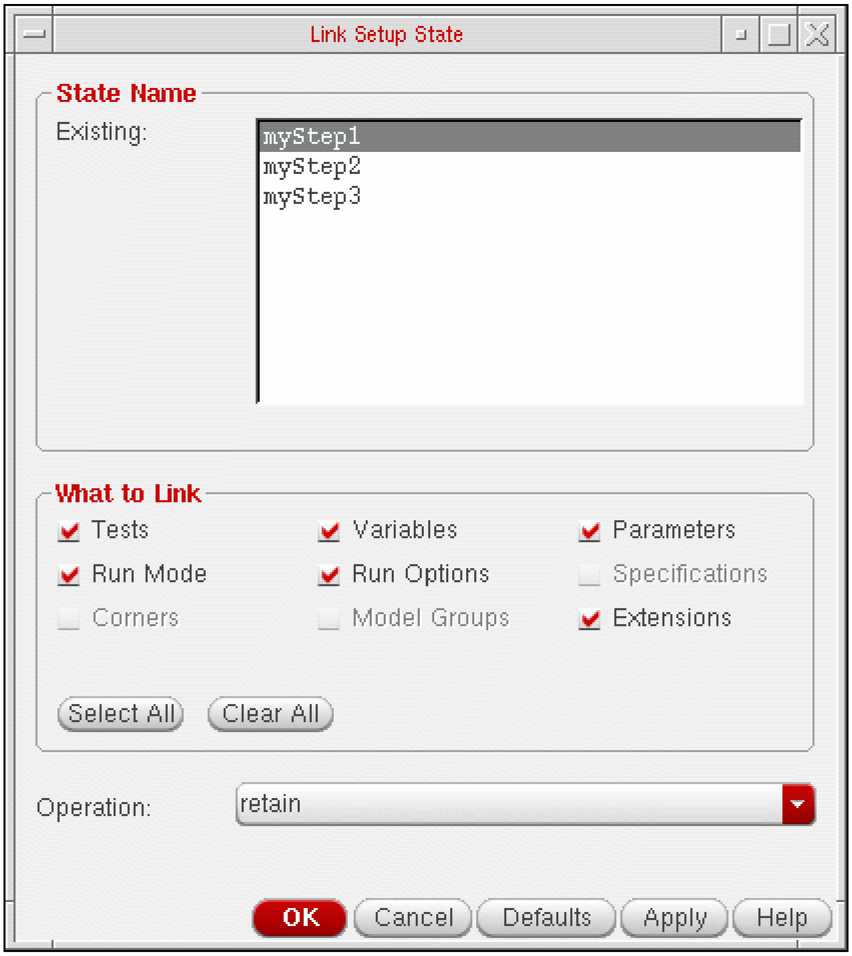
   - To link a setup state to a new OCEAN script file, type the name of the OCEAN script file in the *File Name* field, and click *Open*.
   - To link a setup state to an existing OCEAN script file, select the OCEAN script file, and click *Open.* The following message appears:

     ```
     File <filename> already exists. How do you want to link to it.
     ```

     Do one of the following:

     - Click *Append* to add the OCEAN XL commands required to link the setup state in the end of the file.
     - Click *Overwrite* to delete the contents of the file and add only the OCEAN XL commands required to link the setup state.

The Link Setup State form appears.



4. In the *State Name* group box, select the name of the setup state you want to link to the OCEAN XL script file.

5. In the *What to Link* group box, select the check box next to the settings you want to be restored to the simulation setup in memory.

   By default, all the settings existing in the setup state are selected. You can click *Select All* to select all the settings, or click *Clear All* to clear all the selections.

Select	**To**
Tests	Restore tests
Run Mode	Restore run mode settings

Corners	Restore corners
Variables	Restore global variables
Run Options	Restore run options
Model Groups	Restore model groups
Parameters	Restore parameters
Specifications	Restore specifications
Extensions	Restore extensions

6. In the *Operation* drop-down list, select the restore mode.

Select	**To**
retain	Retain the setup information in memory and append other setup information for each setting selected in the *What to Link* group box. For more information, see How the Simulation Setup is Updated When You Load Setup States on page 791.
merge	Overwrite the setup information in memory for items that have the same name and append other setup information for each setting selected in the *What to Link* group box. For more information, see How the Simulation Setup is Updated When You Load Setup States on page 791.
overwrite	Overwrite the setup information in memory for each setting selected in the *What to Link* group box. For more information, see How the Simulation Setup is Updated When You Load Setup States on page 791.

7. Click *OK* to link the setup state to the OCEAN script file.

The OCEAN script file is opened in a text editor. The OCEAN script file contains the OCEAN XL commands required to required to link the setup state.

```
;----------------- Setup State "myStep1" ---------------
ocnSetXLMode()
ocnxlTargetCellView( "workshopLib" "MSPS_DUT" "adexl" )
ocnxlLoadSetupState( "myStep1" 'retain ?tests t ?vars t ?parameters t
?currentMode t
    ?runOptions t ?extensions t ?specs nil ?corners nil
    ?modelGroups nil )
ocnxlRun()
```

```
ocnxlOutputSummary()
ocnxlEndXLMode()
```

8. Repeat steps 2 to 7 to link more setup states to the OCEAN script file. The OCEAN XL commands required to required to link the setup states are appended in the order in which they are linked to the OCEAN script file.

The following example shows how three setup states named `myTask1`, `mytask2` and `myTask3` are linked to an OCEAN script file.

```
;----------------- Setup State "myStep1" ---------------
ocnSetXLMode()
ocnxlTargetCellView( "workshopLib" "MSPS_DUT" "adexl" )
ocnxlLoadSetupState( "myStep1" 'retain ?tests t ?vars t ?parameters t
?currentMode t
    ?runOptions t ?extensions t ?specs nil ?corners nil
    ?modelGroups nil )
ocnxlRun()
ocnxlOutputSummary()
ocnxlEndXLMode()
;----------------- Setup State "myStep2" ---------------
ocnSetXLMode()
ocnxlTargetCellView( "workshopLib" "MSPS_DUT" "adexl" )
ocnxlLoadSetupState( "myStep2" 'retain ?tests t ?vars t ?parameters t
?currentMode t
    ?runOptions t ?extensions t ?specs nil ?corners nil
    ?modelGroups nil )
ocnxlRun()
ocnxlOutputSummary()
ocnxlEndXLMode()
;----------------- Setup State "myStep3" ---------------
ocnSetXLMode()
ocnxlTargetCellView( "workshopLib" "MSPS_DUT" "adexl" )
ocnxlLoadSetupState( "myStep3" 'overwrite ?tests t ?vars t ?parameters t
?currentMode t
    ?runOptions t ?extensions t ?specs nil ?corners nil
    ?modelGroups nil )
ocnxlRun()
ocnxlOutputSummary()
ocnxlEndXLMode()
```

9. Load and run your OCEAN script file in the Command Interpreter Window (CIW).

For more information, see the OCEAN Reference.

Important

You must not load and run your OCEAN XL script in the CIW while ADE XL is still running for the same cellview.

When you run the OCEAN XL script, the program reports the following information:

- ❑ Sweep parameters and their values

- ❑ Number of tests, sweep points, and corners
- ❑ Points completed and job status information
- ❑ Results location to the output area of the CIW

For example:

```
1/1 completed.
*Info*   The result of this OCEAN XL run are saved in "Interactive.3" in library
"workshopLib", cell "MSPS_DUT", view "adexl".
```

The results location corresponds to the lib/cell/view specified in the `ocnxlTargetCellView` call, such as

```
ocnxlTargetCellView( "workshopLib" "MSPS_DUT" "adexl" )
```

See “OCEAN Commands in XL Mode” in the *OCEAN Reference* for information about OCEAN script commands for ADE XL.

17

Working with Checkpoints

You can use checkpoints in the ADE XL environment to save the active configuration of data such as corners analysis setup, global variables, sweep and test definitions. The environment creates a checkpoint prior to running a simulation and stores checkpoints in your project directory. You can restore fragments of a checkpoint (such as just the sweep setup), edit the data, and run the simulation again. Checkpoints appear as part of history items on the History tab of the Data View assistant pane in your ADE XL environment.

See the following topics for more information:

See also "History Tab Right-Click Menus" on page 831.

Expanding and Collapsing Tree Branches

To expand a tree branch on the History tab of the Data View assistant pane, do the following:

➤ Click *+* to expand an item.

The set of items belonging to that branch appear beneath it.

Tip

Another way to expand a tree branch on the History tab is to right-click the branch and choose *Expand* from the pop-up menu.

To collapse a tree branch, do the following:

➤ Click - to collapse an item.

The set of items belonging to that branch are no longer visible.

Tip

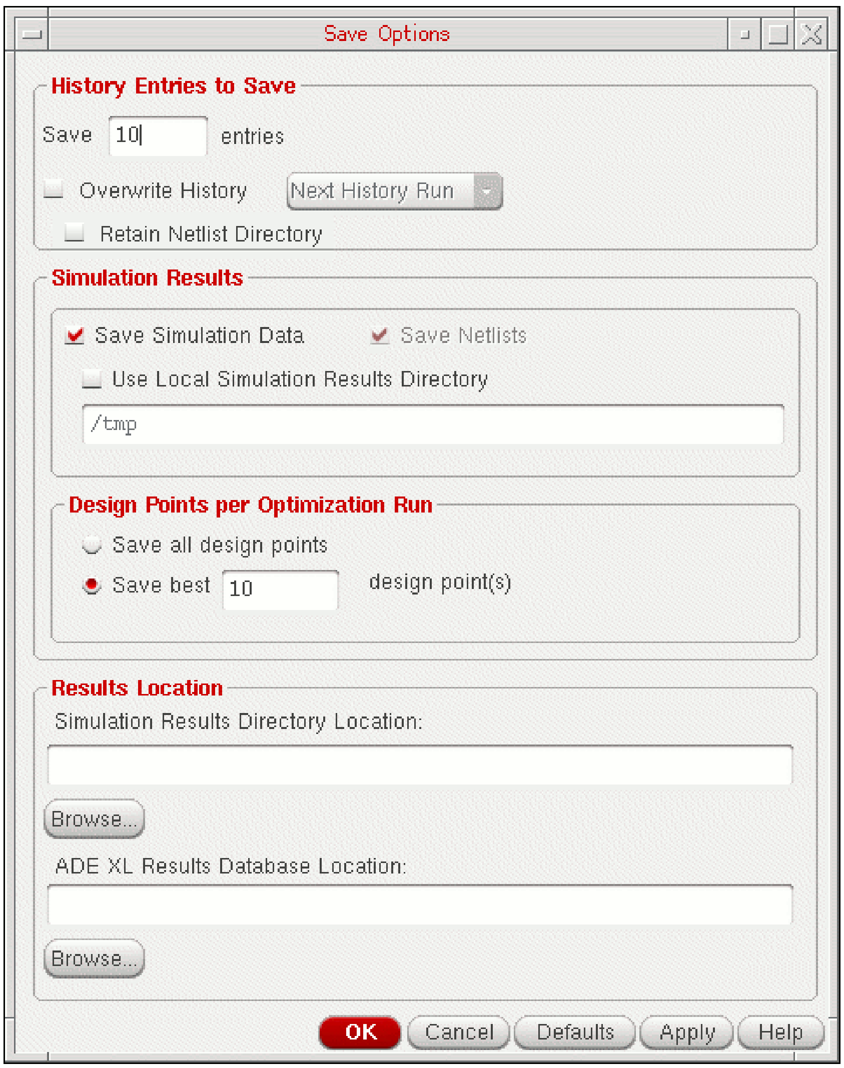
You can also double-click most items to expand or collapse them. The exception is the top branches in the tree: You can type new names for the top branches in the tree by double-clicking them. See Renaming Checkpoints for more information.

Specifying How Much Data to Save

To specify the number of history entries for which you want to save simulation data, do the following:

1. Choose *Options – Save*.

   The Save Options form appears.



2. In the *History Entries to Save* group box, type the number of history items you want to save.

   Note: The *History Entries to Save* group box setting is ignored when you submit a point for evaluation (see Submitting a Point) or troubleshoot a design or data point (see Troubleshooting a Design or Data Point).

3. Click *OK*.

   The program saves only as many history items as the number you typed and automatically removes (does not retain) any unlocked history items above and beyond that number. The reference history item used for incremental simulation runs (see Running an Incremental Simulation) is not automatically removed.

   For information about how history entries are saved when you open an ADE XL view in read-only mode, see Number of History Entries Saved for Read-Only ADE XL Views.

   See Locking and Unlocking a History Item for information about how to lock or unlock history items.

Note: For information on the *Design Points per Optimization Run* group box settings, see "Specifying How Much Optimization Data to Save" in the *Virtuoso Analog Design Environment GXL User Guide*.

Overwriting a History Item during Subsequent Simulation Runs

By default, a new history item is created for each simulation run. If you want to do any of the following, you can specify that the selected history item be overwritten during subsequent simulation runs:

- Save disk space by using the same history item for subsequent simulation runs.
- Reuse the netlist from a history item.
- Reuse the OCEAN and other scripts that you use with your design

  If a new history item is created for each simulation run, the netlist and result directory paths change for each simulation run. This requires you to modify these directory paths in the OCEAN and other scripts that you use with your design for each simulation run. However, when a history item is set to be overwritten, the same netlist and result directories are used for subsequent simulation runs. This enables you to reuse the OCEAN and other scripts because you need not change the directory paths in the scripts.
- Perform any other task that requires that the netlist directory does not change for subsequent simulation runs. Such tasks include:
  - Saving simulation snapshots at specified timepoints during a transient analysis simulation that is run by using the Virtuoso Spectre or Virtuoso UltraSim simulator, and later restarting the simulation from a specific snapshot. For more details about how to save the state of a simulation run at a given timepoint, refer to the Virtuoso *Analog Design Environment L user guide*.
  - Reusing state files created during simulation runs for subsequent simulation runs. For example, you can reuse the `spectre.ic` and `spectre.fc` state files containing DC operating point output at the first and last step of a transient analysis sweep in subsequent simulation runs.

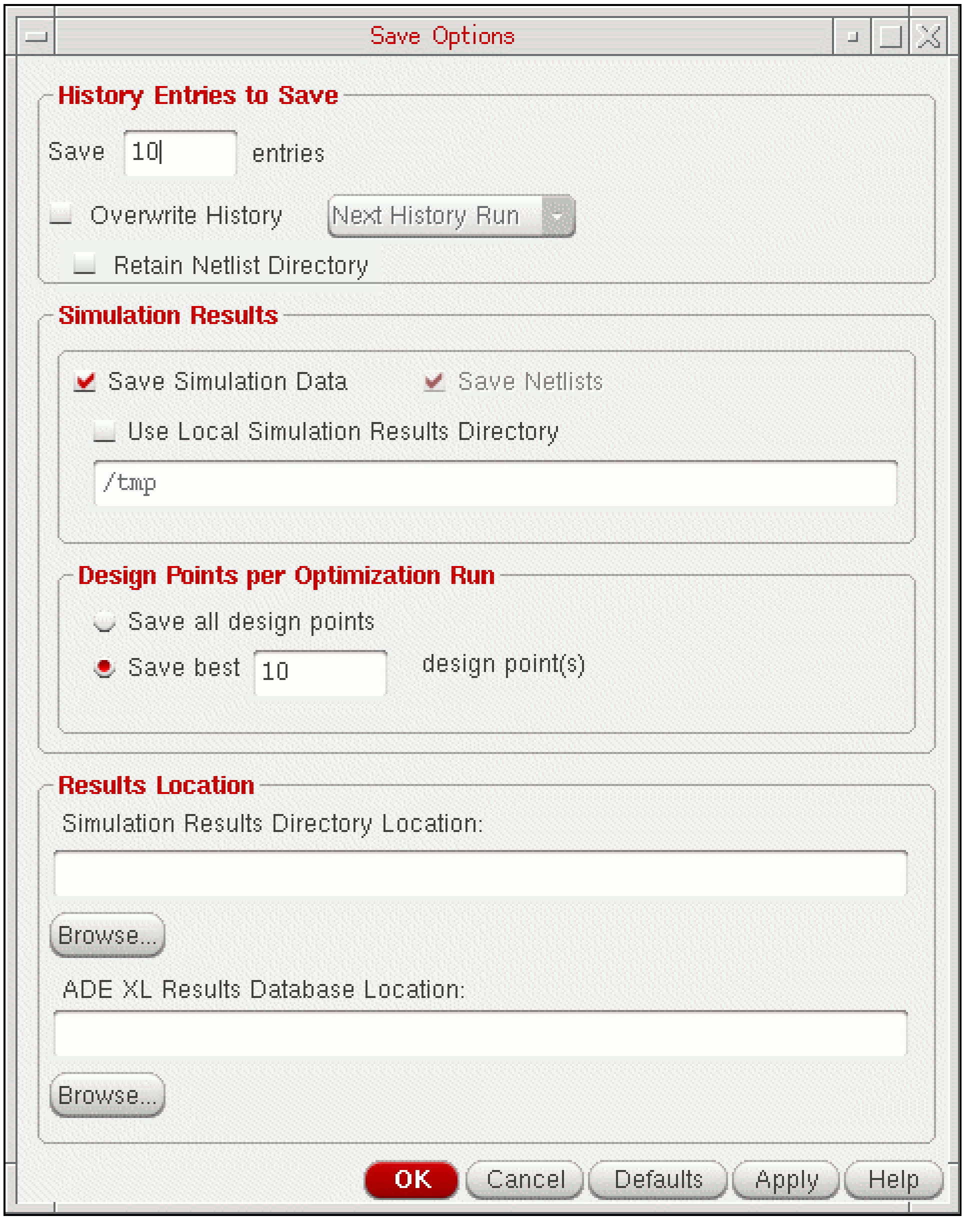
Video

You can view video demonstration for this feature at Overwriting a History Item during Simulation Runs in Virtuoso Analog Design Environment XL.

To overwrite a history item during subsequent simulation runs, do the following:

1. Choose *Options – Save*.

The Save Options form appears.



2. Select the *Overwrite History* check box.

   Note: If the number of histories to be saved is specified as `1` in the *Save* field, the *Overwrite History* check box is automatically selected. The value displayed in the *Save* field is determined by the value of the saveLastNHistoryEntries environment variable. If required, you can edit the value in this field.

3. From the *Overwrite History* drop-down list, select one of the following:

- ❑ Select the history item you want to be overwritten every time a simulation is run.
- ❑ Select *Next History Run* if you want the history item created for the next simulation run to be overwritten during simulation runs.

  For example, if you run Single Run, Corners and Sweeps analysis after selecting *Next History Run*, the history item, say `Interactive.5`, that is created will be automatically selected as the overwrite history item. The `Interactive.5` history item will then be overwritten for subsequent simulation runs.

4. By default, the netlist directory of a previous run history is deleted and then created again before the next run. If you want to reuse the files related to simulation, such as the state files that are created during the previous history runs, you can select the *Retain Netlist Directory* check box. When this check box is selected, the netlist directory of the previous history is retained to reuse the simulation information in the next run. Note that the netlist is generated incrementally in case you change any design information or some of the simulation settings.

   Note: The netlist directory of a history item is retained only if the run mode has not changed after the last simulation run.

5. Click *OK*.

Important Points to Note

- The *Overwrite History* drop-down list displays the names of only the history items that are not locked or selected as a reference history item (see Running an Incremental Simulation on page 497 for more details about reference history items).
- The *Overwrite History* option is automatically set to *Next History Run* if you lock, rename, or delete the history item that was set to be overwritten.
- The *Overwrite History* check box is automatically deselected if an operation on the history item set to be overwritten requires the history item to be referenced. History items are referenced when you do any of the following:
  - ❑ Submit a point (see Submitting a Point)
  - ❑ Simulate only error or incomplete points (see Simulating Only Error or Incomplete Points on page 517)
  - ❑ Troubleshoot a design or data point (see Troubleshooting a Design or Data Point on page 519)
  - ❑ Run incremental simulation (see Running an Incremental Simulation on page 497)

For example, if the history item named `Interactive.5` that is set to be overwritten is also selected as a reference history item for an incremental simulation run, the *Overwrite History* check box is automatically deselected and `Interactive.5` is not overwritten during subsequent simulation runs.

- The values for the *Overwrite History* and *Next History Run* options are saved in the setup database, and the values for the other save options are saved in the ADE XL project directory.
- To apply default values for the two overwrite history options that can be used by all the adexl views, you can use the `axlSetOverwriteHistory` and `axlSetOverwriteHistoryName` SKILL functions in the `.cdsinit` file. For more details on these functions, see *ADE XL SKILL Reference Guide*.

  For more details on saving simulation data by using the Save Options form, see Specifying Options for Saving Simulation Results.

Viewing Active Setup Details

To view details for the active setup, do the following:

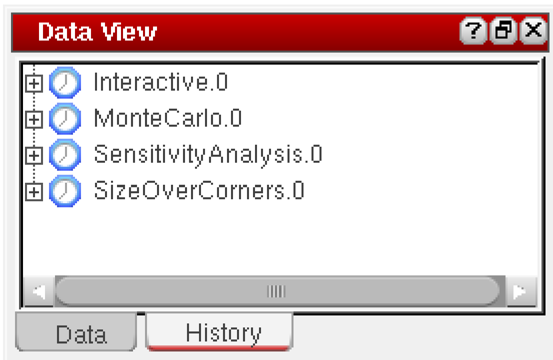
1. On the Setup DB Viewer assistant pane, click *+* to expand the *Active Setup* tree.

   Details for the current setup appear in the expanded tree.
2. (Optional) Click *+* at each level of the tree to expand it.

Viewing Checkpoints

To view checkpoints, do the following:

➤ Click the History tab of the Data View assistant pane.



The name of the history item is

runType.seqNum

Where, *runType* is one of the following:

- `Interactive` (for Single Run, Sweeps and Corners runs)
- `MonteCarlo` (for Monte Carlo Sampling runs)
- `GlobalOpt` (for Global Optimization runs)
- `LocalOpt` (for Local Optimization runs)
- `SensitivityAnalysis` (for Sensitivity Analysis runs)
- `ImproveYield` (for runs)
- `FeasibilityAnalysis` (for Feasibility Analysis runs)
- `SizeOverCorners` (for Size Over Corners runs)

and *seqNum* is `0` (zero) for the first item, then 1+(the largest existing *seqNum* for that *runType*).

When you troubleshoot a design or data point (see Troubleshooting a Design or Data Point on page 519, the name of the resulting history item is *runType.seqNum.TS.seqNum*

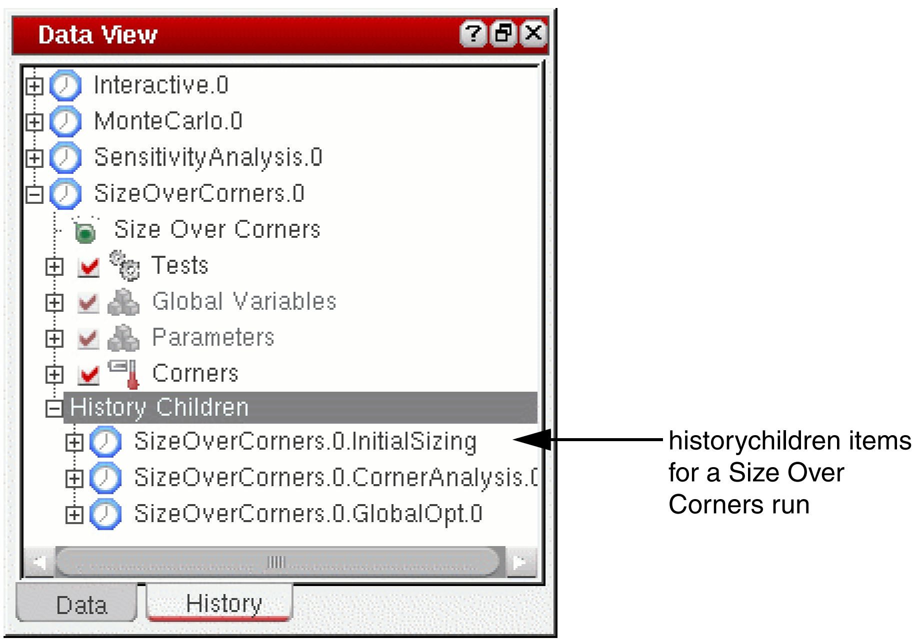
To view details for a particular checkpoint, do the following:

1. On the History tab of the Data View assistant pane, click + to expand the checkpoint whose details you want to view.

   Checkpoint details appear in the expanded branch, including what testbenches were run, what parameters were varied, and any specifications or corners.

2. (Optional) Click + to the left of each level of the tree to expand it.

Note: *ImproveYield*, *Manual Tuning*, and *SizeOverCorners* history items are slightly different from other history items in that they have *historychildren* items that correspond to other simulations that are run during these runs. The *historychildren* items are displayed under the *History Children* tree for a history item, as shown in the figure below.

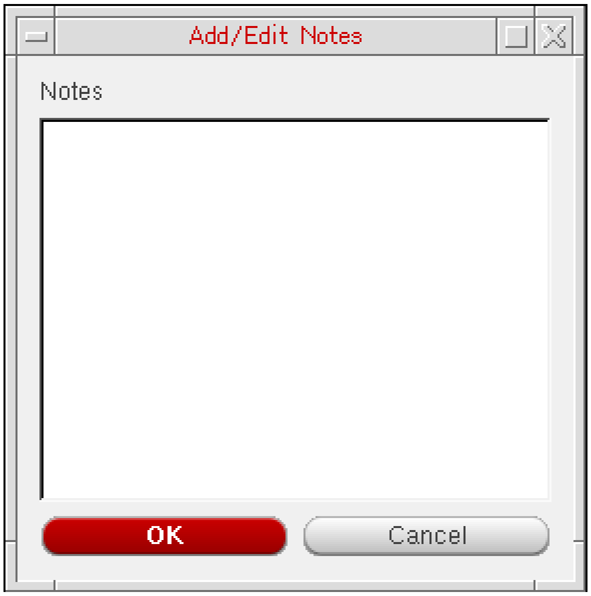


Adding Notes to a Checkpoint

To add notes to a checkpoint, do the following:

1. On the *History* tab of the Data View assistant pane, right-click the history for which you want to add notes and choose *Notes*.

The Add/Edit Notes form is displayed.



2. In the *Notes* field, add notes for the history.

   Note: By default, the notes field can accept only 512 characters. To change the default maximum characters limit, you can set the `maxNotesLength` environment variable.

3. Click *OK*.

For related information, see Adding Notes to a Test.

Renaming Checkpoints

When the program creates a checkpoint, the checkpoint appears in the History tab of the Data View assistant pane.

To rename a checkpoint, do the following:

1. Double-click the checkpoint name (for example, *Interactive.0*).

   The name becomes editable.

2. Type a new name or click-drag to highlight the portion of the name you want to change and type new text.

   The name you type must be unique (that is, not the same as any other checkpoint name).

3. Press *Return* when finished.

   The new name appears in place of the old name in the History tab.

Note the following:

- If you rename *ImproveYield* and *SizeOverCorners* history items, the corresponding *historychildren* items are also automatically renamed.
- You cannot rename *historychildren* items.

Restoring a Checkpoint

You can restore all or parts of a checkpoint.

Restoring an Entire Checkpoint

To restore an entire checkpoint, do the following:

➤ In the History tab of the Data View assistant pane, right-click the checkpoint you want to restore and choose *Load Setup to Active*.

The program restores test setup (including run mode) and parameter information to the various assistant panes. You can access results on the Results tab of the Outputs pane. The results database context in Calculator and Results Browser is also set to the results of this history. See Chapter 14, "Viewing, Plotting, and Printing Results" for more information.

Restoring Part of a Checkpoint

To restore part of a checkpoint, do the following:

1. In the History tab of the Data View assistant pane, expand the history item.
2. In the expanded history item, right-click the item or set of selected items you want to restore and choose *Load Setup to Active*.

   For example, you can

   ❑ Right-click the *Tests*, *Global Variables, Parameters* or *Corners* branch and choose *Load Setup to Active* to restore all tests, corners, or specifications in the checkpoint.

   ❑ Select/highlight one or more:

   ❍ tests under the *Tests* branch, or

   ❍ corners under the *Corners* branch.

   then right-click and choose *Load Setup to Active* to restore them.

   The program restores the selected item or items.

See also "Viewing Results from a Particular Checkpoint" next.

Viewing Results from a Particular Checkpoint

To view the results from a particular checkpoint, do the following:

➤ In the History tab of the Data View assistant pane, right-click the checkpoint whose results you want to view and choose *View Results*.

The results appear on a new tab in the Results tab of the Outputs pane. The name of the tab matches the name of the history item whose results you are viewing.

Important

If you want to access results from a previous checkpoint, you need to restore the checkpoint.

See Chapter 14, "Viewing, Plotting, and Printing Results" for more information.

Saving Results from a Particular Checkpoint

To save the results from a particular checkpoint, do the following:

1. In the History tab of the Data View assistant pane, right-click the checkpoint whose results you want to view and choose *Save Results*.

   The Save Results form appears.

2. Enter the path to the directory where you want to save the results.

   Alternatively click the browse button to specify the directory.

3. (Optional) Select the *Copy PSF Results?* check box if you want to copy PSF results

4. Click *OK*.

Viewing Results for a Particular Checkpoint in the Results Browser Window

To view the results for a particular checkpoint in the Results Browser window, do the following:

➤ In the History tab of the Data View assistant pane, right-click the checkpoint whose results you want to view and choose *Results Browser*.

The Results Browser window appears.

Viewing the Run Log for a Particular Checkpoint

To view the simulation run log file for a particular checkpoint, do one of the following:

➤ Click on the *Open Run Log*  button in the Results tab.

➤ In the History tab of the Data View pane, right-click the checkpoint whose run log file you want to view and choose *Open Run Log*.

The Run Log | Log File Viewer form appears displaying the simulation run log file.

Tip

You can also view the contents of the run log file in the Status view on the Results tab.

Opening a Terminal Window in the Results Directory for a Particular History Item

To open a terminal window in the results directory for a particular history item, do the following:

➤ In the History tab of the Data View assistant pane, right-click the history item whose results directory you want to view and choose *Open Terminal*.

A terminal window opens in the `results/data/<history_item>` directory.
For example, if the history item you right-click is *Interactive.2*, the terminal window opens in the `results/data/Interactive.2` directory.

For more information about the location of the results directory, see Specifying Options for Saving Simulation Results on page 73.

Note: You can specify the command the program uses to open a terminal window using the `adexl.gui openTerminalCommand` environment variable.

ImproveYield and *SizeOverCorners* history items are slightly different from other history items in that the *Open Terminal* menu item appears on the *historychildren* items (such as *ImproveYield.0.GlobalOpt.0*) rather than on the history item itself.

To open a terminal window in the results directory for a particular *ImproveYield* history item, do the following:

➤ In the History tab of the Data View assistant pane, right-click the history children item whose results directory you want to view and choose *Open Terminal*.

A terminal window opens in the `results/data/`*`historyChild`* directory. For example, if the *historychildren* item you right-click is *ImproveYield.0.GlobalOpt.0*, the terminal window opens in the `results/data/ImproveYield.0.GlobalOpt.0` directory.

Deleting the Simulation Data for a History Item

To delete the simulation data for a history item, do the following:

➤ In the History tab, right-click the history item for which you want to delete the simulation data and choose *Delete Simulation Data*.

Deleting a Checkpoint

To delete a checkpoint from the History tab of the Data View assistant pane, do the following:

➤ In the History tab, right-click the history item that contains the checkpoint you want to delete and choose *Delete*.

The program removes the history item and its checkpoint from the *History* tree and the raw simulation results from the `results` directory.

Locking and Unlocking a History Item

To lock a history item so that you cannot delete it, do the following:

➤ In the History tab of the Data View assistant pane, right-click the history item you want to protect from accidental deletion and choose *Lock*.

The  icon that appears to the left of the history item in the History tab becomes a lock.

You cannot delete this item while the lock icon is present.
(The *Delete* item is not available on the right-click pop-up menu.)

The number of locked items does not count toward number of saved items.

To unlock a history item, do the following:

➤ In the History tab of the Data View assistant pane, right-click the history item you want to unlock and choose *Unlock*.

The program replaces the lock icon with the clock icon and you can delete this history item. (The *Delete* item is now available on the right-click pop-up menu.)

Note the following:

- If a history item is locked, all the *historychidren* items listed under the *History Children* tree are also locked. You cannot unlock a *historychidren* item when the history item is locked.
- If a history item is unlocked, all the *historychidren* items listed under the *History Children* tree are also unlocked.
- If you lock a *historychidren* item when the history item is unlocked, the history item and all other *historychidren* items are also locked.

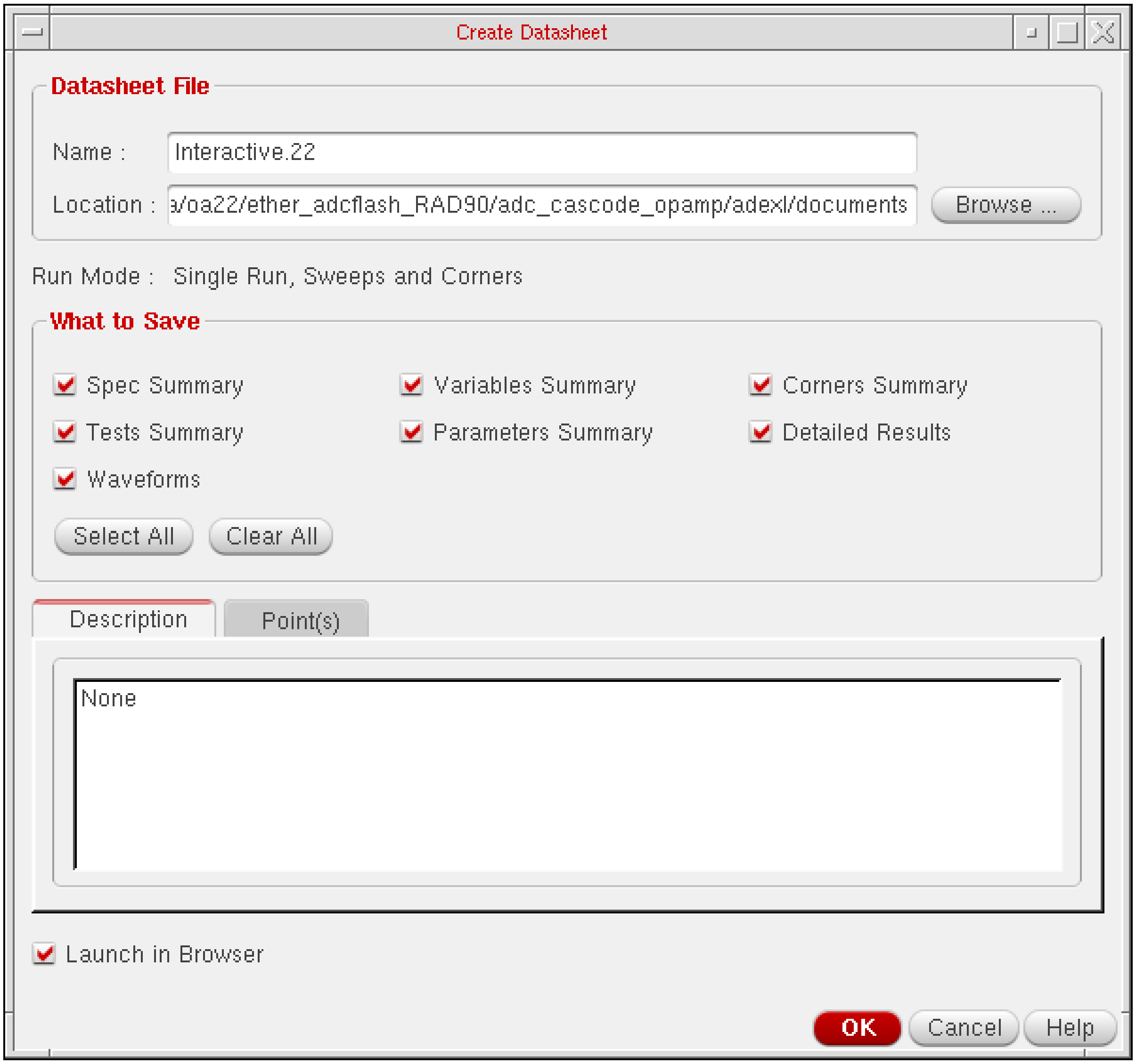
Working with Datasheets

Creating a Datasheet for a Checkpoint

To create a datasheet for a checkpoint, do the following:

1. In the History tab of the `Data View` pane, right-click the checkpoint for which you want to create a datasheet and choose *Create Datasheet*.

The Create Datasheet form appears.



Note the following:

- You can also choose *Create – Datasheet* or click the  toolbar button to open the Create Datasheet form.
- You cannot create a datasheet for a check point when simulations are in progress for that checkpoint.

2. In the *Name* field, type a name for the datasheet.

Note: By default, the name of the datasheet is the name of the checkpoint for which you are creating the datasheet. For example, if you are creating a datasheet for a checkpoint named `Interactive.17`, the default name of the datasheet will be `Interactive.17`.

3. In the *Location* field, type the directory path where you want the program to write your datasheet; or do the following:

   a. Click *Browse*.

   b. On the form that appears, navigate to and select the directory where you want the program to write your datasheet.

   c. Click *Open*.

   Note the following:

   ❑ By default, the datasheet is created in the `documents` folder of the ADE XL view.

   ❑ If you do not specify the absolute path to a directory in the *Location* field, the directory will be created in your current working directory (the directory in which you ran the `virtuoso` command). For example, if you run the `virtuoso` command from `/net/designs/myDesign` and type `myDatasheets` in the *Location* field, datasheets will be created in the `/net/designs/myDesign/myDatasheets` directory.

4. (Optional) Select or deselect the following check boxes:

Check Box	Description
Spec Summary	When turned on, this option writes spec summary information to the datasheet. Spec summary information is displayed in the following sections in the datasheet: ■ *Results Summary* section in the main datasheet page ■ *Results Summary* section in the datasheet page for each test The text *Various* in the *Target* column in the *Results Summary* section indicates that you have overridden or disabled one or more corner specifications for a measurement. For more information, see Disabling and Enabling Corner Specifications on page 710. For information about the colors used in the *Minimum Value* and *Maximum Value* columns in the *Results Summary* section, see Colors Used to Display Status of Measured Values and Specifications in Datasheets on page 828.
Variables Summary	When turned on, this option writes variables summary information to the datasheet, including names and values of variables. Variables summary information is displayed in the *Variables* section in the main datasheet page.
Corners Summary	When turned on, this option writes corners summary information to the datasheet, including the list of corners enabled or disabled for each test, values of parameters and design variables at corners, and the model groups added for each corner. Corners summary information is displayed in the *Corners* section in the main datasheet page.

Check Box	Description
Tests summary	When turned on, this option writes test summary information to the datasheet, including test name, test design, simulator, and state information. Tests summary information is displayed in the *Tests Summary* section in the main datasheet page.
Parameters Summary	When turned on, this option writes parameters summary information to the datasheet, including names and values of parameters. Parameters summary information is displayed in the *Parameters* section in the main datasheet page.
Detailed Results	When turned on, this option writes detailed results information for each test to the datasheet, including parameter values for each run (sweeps, corners) and output values for each measurement expression. Detailed results information for each test is displayed in the *Detailed Results* section in the datasheet page for that test. For information about the colors used in the *Value* and *Target* columns in the *Outputs* sub-section of the *Detailed Results* section, see Colors Used to Display Status of Measured Values and Specifications in Datasheets on page 828. **Note:** The *Target* column is displayed for a measurement in the *Outputs* sub-section of the *Detailed Results* section only if you have overridden or disabled a corner specification for the measurement. For more information about overriding or disabling corner specifications, see Working with Specifications on page 702.

Check Box	Description
Waveforms	When turned on, this option displays waveform images in the datasheet files for each test. The waveform images display the outputs for which the *Plot* check box is selected in the Outputs Setup tab of the Outputs pane. If you want only specific results to be displayed in the waveform images in a datasheet, do the following before you create the datasheet: **1.** On the Outputs Setup tab of the Outputs pane, select the *Plot* check box next to the outputs that you want to include in the waveforms. **2.** On the Results tab of the Outputs pane, click the button. The waveform images are saved as Portable Network Graphic (`.png`) image files in the datasheet directory. For more information about the location of the datasheet directory, see Location of Datasheet Files on page 827. **Note:** By default, the waveform images in the results directory for a history item are displayed in the datasheet. If waveform images are not available for a history item, they are automatically created and displayed in the datasheet. See also: Displaying Customized Waveform Images in the Data Sheet on page 829

Check Box	Description
Launch in browser	When turned on, this option launches the datasheet in an web browser. The datasheet is displayed in the browser specified in the *Web Browser* field of the CIW's User Preferences form. The default value of the *Web Browser* field is `netscape`. You can also specify the default browser to be used by setting the following environment variable in the your `.cdsenv` file before starting the Virtuoso Design Environment: `ui webBrowser string "`*`browserName`*`"` where `browserName` is the name of the browser's executable. The default value for this environment variable is `netscape`. **Note:** Ensure that your web browser supports XML and XSLT.

5. (Optional) In the Points tab, do one of the following:

Select	To
Design Point ID	Include results information only for a specific design point in the datasheet. Type the ID of the design point in the *Design Point ID* field. **Note:** You can find the ID for a design point from the *Point* column in the Results tab of the Outputs pane.
All Design Points	Include results information for all the design points in the datasheet.

6. (Optional) In the Description tab, enter a description for the datasheet.
7. Click *OK*.

The program writes the datasheet information you requested. If you selected the *Launch in browser* check box, the program also launches the datasheet in an HTML browser.

Location of Datasheet Files

The datasheet files are saved in a directory that has the same name as the name of the datasheet. For example, if you specified the name of the datasheet as `myDatasheet` in the Create Datasheet form, the datasheet files are saved in the `myDatasheet` directory in the location specified in the Create Datasheet form.

The datasheet directory contains the following files:

Table 17-1 Datasheet Files

File Name	Description
`<`*`datasheet_name`*`>.xml`	The main datasheet file. For example, if you specified the name of the datasheet as `myDatasheet` in the *Name* field, the main datasheet file will have the name `myDatasheet.xml`.
`<`*`datasheet_name`*`>_<`*`test_name`*`>.xml`	Datasheet file for a test that is created if the *Detailed Results* or *Waveforms* check boxes are selected. For example, if you specified the name of the datasheet as `myDatasheet` in the *Name* field, and you have two tests named `AC1` and `TRAN1`, the following two datasheet files that contain the detailed results information for each test are created if the *Detailed Results* or *Waveforms* check boxes are selected: ■ `myDatasheet_AC1.xml` ■ `myDatasheet_TRAN1.xml`
`dsImg*.png`	Portable Network Graphic (`.png`) files for the waveform images displayed in datasheets.

Colors Used to Display Status of Measured Values and Specifications in Datasheets

The following colors are used as the background color for cells in the *Minimum Value* and *Maximum Value* columns in the *Results Summary* section to display the status of measured values:

Cell Background Color	Description
Green	Indicates a pass status—all the measured values are within the limits defined by the specification.
Yellow	Indicates a near status—one or more measured values are no more than 10% outside the target value of the specification.
Red	Indicates a fail status—one or more measured values are greater than 10% outside the target value of the specification.

The following colors are used as the background color for cells in the *Value* and *Target* columns in the *Outputs* sub-section in the *Detailed Results* section to display the status of measured values and corner specifications:

Cell Background Color	Description
Green (in *Value* column)	Indicates a pass status for the measured value—the measured value is within the limits defined by the specification.
Yellow (in *Value* column)	Indicates a near status for the measured value—the measured value is no more than 10% outside the target value of the specification.
Red (in *Value* column)	Indicates a fail status for the measured value—the measured value is greater than 10% outside the target value of the specification.
Green (in *Target* column)	Indicates a pass status for the corner specification—the measured value is within the limits defined by the specification.
Yellow (in *Target* column)	Indicates a near status for the corner specification—the measured value is no more than 10% outside the target value of the specification.
Red (in *Target* column)	Indicates a fail status for the corner specification—the measured value is greater than 10% outside the target value of the specification.

Cell Background Color	Description
Grey (in *Target* column)	Indicates that a corner specification is disabled.

Displaying Customized Waveform Images in the Data Sheet

If the setup in a history item is the same as the active setup, you can display customized waveform images for the history item by doing the following:

1. On the Results tab of the Outputs pane, click the  button.

   Waveforms are displayed in waveform windows.

2. Customize the waveforms as required in the waveform window.

3. Create the datasheet without closing the waveform windows in which you customized the waveforms.

The customized waveforms are displayed in the data sheet.

Opening a Datasheet

To open a datasheet, do one of the following:

➤ In the *Documents* tree on the Data View pane, double-click on the HTML file for the datasheet.

  The HTML file for the datasheet has the same name as the name of the datasheet. For example, if you specified the name of the datasheet as `myDatasheet` in the Create Datasheet form, the HTML file for the datasheet will have the name `myDatasheet.html`. For more information about working with documents, see Chapter 18, “Working with Documents.”

➤ Open the `.xml` files in the datasheet directory in a web browser.

  For more information about the `.xml` files in the datasheet directory, see Location of Datasheet Files on page 827.

Customizing the Datasheet Format and Structure

The following stylesheet files are used to control the format and structure of datasheets. These files are located at <*your_install_dir*>`/share/cdssetup/adexl`.

File Name	Description
datasheetMainDoc.xsl	XSLT stylesheet that controls the structure of the main datasheet file.
datasheetTestDoc.xsl	XSLT stylesheet that controls the structure of the datasheet files that contain the detailed results information for each test.
datasheet.css	CSS stylesheet that controls the formatting of the main datasheet file and the datasheet files that contain the results information for each test.

To customize the structure of the main datasheet page, do the following:

1. Copy the `datasheetMainDoc.xsl` file in the <*your_install_dir*>`/share/cdssetup/adexl` directory to a different location.

   Rename the file, if required.

2. Open the file in a text editor and make the required changes.

3. Use the `adexl.datasheet mainDocXSLFile` environment variable to specify the location of the file. For more information about this environment variable, see mainDocXSLFile on page 932.

To customize the structure of the datasheet pages for tests, do the following:

1. Copy the `datasheetTestDoc.xsl` file in the <*your_install_dir*>`/share/cdssetup/adexl` directory to a different location.

   Rename the file, if required.

2. Open the file in a text editor and make the required changes.

3. Use the `adexl.datasheet testDocXSLFile` environment variable to specify the location of the file. For more information about this environment variable, see testDocXSLFile on page 932.

To customize the formatting of the datasheet files, do the following:

1. Copy the `datasheet.css` file in the `<your_install_dir>/share/cdssetup/adexl` directory to a different location.

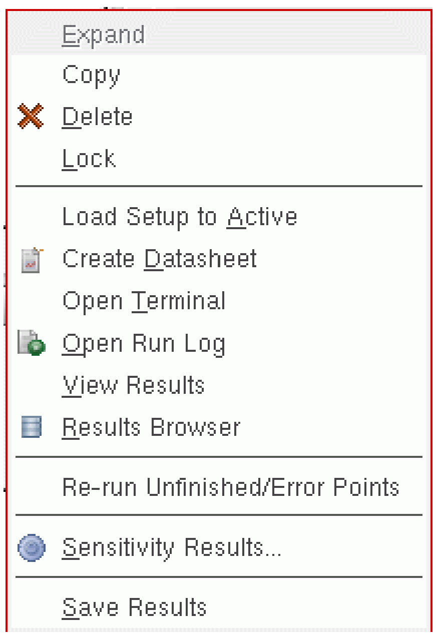
   Rename the file, if required.

2. Open the file in a text editor and make the required changes.

3. Use the `adexl.datasheet CSSFile` environment variable to specify the location of the file. For more information about this environment variable, see CSSFile on page 931.

History Tab Right-Click Menus

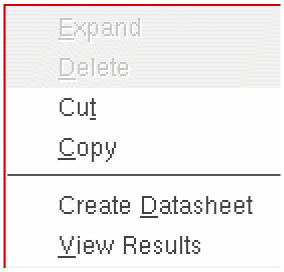
When you right-click a history item in the History tab of the Data View assistant pane, the following pop-up menu appears:



Note: For *ImproveYield* history items, this menu appears when you right-click a *historychildren* item (such as *ImproveYield.0.GlobalOpt.0*).

When you right-click a checkpoint within a history item, the pop-up menu is the same as the one shown above without the *Lock* entry.

When you right-click the results item for a checkpoint in the History tab of the Data View assistant pane, a pop-up menu similar to the following appears:



You can select appropriate menu actions for the item you right-clicked.

18

Working with Documents

ADE XL allows you to add documents that you want to refer to when working with the design. For example, you can add the design specifications document for your design, so that you can refer to it when working with the design.

The added documents are displayed in the *Documents* folder in the Data View pane. You can double-click on a document to open it in its associated program.

See the following topics for more information:

- Adding Documents on page 833
- Opening Documents on page 834
- Removing Documents on page 834

Adding Documents

You can add documents in the HTML, CSV, PDF and text documents in ADE XL.

To add a document, do the following:

1. Choose *Create – Document*.

Tip

Alternatively, you can click where it says *Click to add document* in the Document folder on the Data View assistant pane.

The Choose Documents to be Added form appears.

2. Select the document you want to add and click *Open*.

The document is displayed in the *Documents* folder in the Data View pane.

Important

The documents you add are copied to your ADE XL view. If you update a document at its original location, ensure that the latest version of the document is added in ADE XL.

Opening Documents

To open a document, do the following:

➤ In the *Documents* tree on the Data View pane, double-click on the document.

The document is opened in its associated program. For example, if you open a PDF file, it is opened in Adobe Acrobat.

Removing Documents

Caution

There is no undo for this action.

To remove a document, do the following:

➤ In the *Documents* tree on the Data View pane, right-click the document you want to remove and choose *Delete*.

The program deletes the selected document.

Note: When a document is removed, it is only deleted from your ADE XL view. It is not deleted from the location from which you added the document.

Saving Documents

You can save a document displayed in the *Documents* tree on the Data View pane under a different name or at a different location.

To save a document under a different name or at a different location, do the following:

1. In the *Documents* tree on the Data View pane, right-click the document you want to save under a different name or at a different location and choose *Save As*.

   The Save Document As form appears.

2. Select the directory in which you want to save the file.
3. In the *File name* field, enter the name under which you want to save the document.
4. Click *Save*.

A

Environment Variables

This appendix describes the public environment variables that control the characteristics of the ADE XL and GXL Environment. You can customize the operation and behavior of ADE XL and GXL features and forms by changing the values of particular environment variables. The default value of each variable appears in the syntax descriptions.

See the following sections for more information:

adexl.setupdb

loadSetupToActiveAlsoViewsResults

Specifies if the *Load Setup To Active* command should display the results in addition to loading the setup details from a history.

By default, ADE XL loads the results of a history while loading the setup details. When the results are large, loading them takes a lot of time. Setting this variable to `nil` loads only the setup details.

In `.cdsenv`:

```
adexl.setupdb loadSetupToActiveAlsoViewsResults boolean t
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.setupdb" "loadSetupToActiveAlsoViewsResults" 'boolean
     nil)
```

Valid Values:

`t`	The *Load Setup To Active* command displays the results in addition to loading the setup details from a history.
`nil`	The *Load Setup To Active* command only loads the setup details from a history and does not show results.

Default Value: `t`

saveDir

Specifies where you want the program to write the setup database file.

Note: If you do not specify a `saveDir`, or if the `saveDir` you specify is not valid, the program writes the setup database file to the ADE XL view. If your design library is set up as read-only, you can use this environment variable to specify a writable location.

In `.cdsenv`:

```
adexl.setupdb saveDir string ""
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.setupdb" "saveDir" 'string "" )
```

Valid Values:

Any valid directory path

percentageForNearSpec

Specifies the percentage value based on which the *near* status is displayed in the *Pass/Fail* column on the Results tab of the Outputs pane when one or more measured values for an output are no more than the percentage value outside the target value of the specification.

For more information about the *near* status in the *Pass/Fail* column on the Results tab, see Viewing Specification Results in the Results Tab on page 737.

In `.cdsenv`:

```
adexl.setupdb percentageForNearSpec int 10
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.setupdb" "percentageForNearSpec" 'int 10 )
```

Valid Values:

Any integer between than `0` and `99`

Note: If set to `0`, only the *pass* or *fail* status is displayed in the *Pass/Fail* column on the Results tab.

Default Value: `10`

useNMPForMapping

Specifies whether nmp-based name mapping scheme must be used for naming files created by ADE XL and ADE GXL.

Important

Cadence recommends setting this environment variable to `t` if you are using a design management system.

In `.cdsenv`:

```
adexl.setupdb useNMPForMapping boolean t
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.setupdb" "useNMPForMapping" 'boolean t)
```

Valid Values:

`t`	Uses nmp-based name mapping scheme for naming files. **Note:** Only the files in views that were created when this variable is set to `t` will have names assigned using the nmp-based name mapping scheme. Files in views that were created when this variable is not set or set to `nil`, will continue to be named using the default name mapping scheme.
`nil`	Uses the default name mapping scheme for naming files.

Default Value: `nil`

adexl.test

autoCopyCellviewVars

Controls copying of new design variables and new values for existing design variables from the design associated with a test when you open an ADE XL view or add a test in ADE XL.

In `.cdsenv`:

```
adexl.test autoCopyCellviewVars boolean t
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.test" "autoCopyCellviewVars" 'boolean t)
```

Valid Values:

`t`	Automatically copies design variables from the design associated with a test when you open an ADE XL view or add tests in an ADE XL view.
`nil`	Disables the automatic copy of design variables when you open an ADE XL view or add tests in an ADE XL view.

You can do one of the following to manually copy new design variables and new values for existing design variables from the design associated with a test:

- On the Variables tab of the Variables and Parameters pane, right-click the test and choose *Copy from Cellview*.
- In an expanded test tree on the Data View pane, right-click a design variable and choose *Copy from Cellview*.

Default Value:	`nil`

autoPromoteVarsToGlobal

Controls whether design variables are automatically added as global variables on the Data View and on the Variables tab of the Variables and Parameters pane.

In `.cdsenv`:

```
adexl.test autoPromoteVarsToGlobal boolean t
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.test" "autoPromoteVarsToGlobal" 'boolean t)
```

Valid Values:

`t`	All design variables are automatically added as global variables in the *Global Variables* tree on the Data View and the Variables tab of the Variables and Parameters pane.
`nil`	Disables the automatic addition of design variables as global variables.

Default Value:	`t`

checkForUnsavedViewsUponRun

Controls whether unsaved design views should be checked before running simulations.

In `.cdsenv`:

```
adexl.test checkForUnsavedViewsUponRun boolean t
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.test" "checkForUnsavedViewsUponRun" 'boolean t)
```

Valid Values:

`t`	Checks for unsaved design views before running simulations.
`nil`	Does not check for unsaved design views before running simulations. You can choose this option for better performance.

Default Value: `t`

debugDataDir

Controls where to save results for the simulations run from ADE XL Test Editor.

In `.cdsenv`:

```
adexl.test debugDataDir boolean t
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.test" "debugDataDir" 'string "./debugResults")
```

Valid Values:

Any valid directory path

Default Value: `nil`

initiallyAddNameUniqifier

Appends a sequence number to the end of test name to make it unique.

When you create a new test, ADE XL provides a name to the test by using a default format or by using the test name returned by the axlCustomADETestName API, if defined. If the *initiallyAddNameUniqifier* environment variable is set to `t`, the tool appends a sequence number to it to make it unique.

In `.cdsenv`:

```
adexl.test initiallyAddNameUniqifier boolean t
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.test" "initiallyAddNameUniqifier" 'boolean nil
```

Valid Values:

`t`	Appends a unique number to the test name.
`nil`	Does not append a unique number to the test name.

Default Value: `t`

adexl.simulation

autoDetectNetlistProcs

Controls whether cellviews that use netlist procedures are automatically detected and renetlisted every time the design is netlisted.

In `.cdsenv`:

```
adexl.simulation autoDetectNetlistProcs boolean t
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.simulation" "autoDetectNetlistProcs" 'boolean t)
```

Valid Values:

`t`	Automatically detects cellviews that use netlist procedures and renetlists these cellviews every time the design is netlisted.
`nil`	Disables the automatic-detection of cellviews that use netlist procedures. These cellviews will not be renetlisted every time the design is netlisted. **Note:** Even if the value is set to `nil`, a cellview that uses netlist procedures is renetlisted if parameters are specified in the *Parameters* tab of the Variables and Parameters pane for an instance of the cellview.

Default Value: `nil`

See also:

- overrideNetlistProcDetection on page 855

createCompositeSimLogFileWhenSimCountFewerThan

By default, ADE XL creates a composite output log if there are upto `100` points for which simulations are to be run. However, if the number of data points is very large, a lot of disk space and time is taken to create and save the composite log. In such cases, you can use this variable to specify the maximum number of simulations up to which ADE XL should save a composite output log file for the outputs.

Note: If the composite output log file is not saved, the *Output Log* command in the context-sensitive menu for an output is not enabled.

In `.cdsenv`:

```
adexl.simulation createCompositeSimLogFileWhenSimCountFewerThan int
     50
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.simulation"
     "createCompositeSimLogFileWhenSimCountFewerThan" 'int 45)
```

Valid Values:

A positive integer value between `0` and `100000`

Default Value: `101`

createRunLogForSweepsCorners

Specifies if the run log created for the *Single Run, Sweeps and Corners* run mode needs to include the details about the best design point. Adding this information in the run log takes time. Therefore, by default, ADE XL writes the best design point for this run mode only in the following two scenarios:

- When the *Single Run, Sweeps and Corners* simulation is run as part of the *Manual Tuning* run mode that aims at finding the best design point.
- When the number of points in the *Single Run, Sweeps and Corners* run mode is less than the limit specified by the `createRunLogWhenSimsFewerThan` environment variable.

However, if required, you can choose to include this information in other scenarios as well. For this, set this variable to one of the valid values listed in the table given below.

In `.cdsenv`:

```
adexl.simulation createRunLogForSweepsCorners cyclic
     "WhenMultipleDesignPoints"
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.simulation" "createRunLogForSweepsCorners"
     'cyclic "WhenMultipleDesignPoints")
```

Valid Values:

`Always`	Always appends the details of the best design point in the run log for *Single Run, Sweeps and Corners* run mode.
`ManualTuningOrSimLimited`	Appends the details of the best design point only when the *Single Run, Sweeps and Corners* run mode is run as part of the Manual Tuning run mode or when the number of points is less than the count specified by `createRunLogWhenSimsFewerThan`.
`WhenMultipleDesignPoints`	Appends the details of the best design point only when the simulation includes multiple design points. This information is not added to the run log for a simulation with a single design point.
`SimLimited`	Appends the details of the best design point if the number of points is less than the count specified by `createRunLogWhenSimsFewerThan`.
`OnlyInManualTuning`	Appends the details of the best design point only when the *Single Run, Sweeps and Corners* run mode is run as part of the Manual Tuning run.
`Never`	Never adds the details of the best design point to the run log for the *Single Run, Sweeps and Corners* run mode.

Default Value: `ManualTuningOrSimLimited`

See also:

- Viewing the Run Log for a Particular Checkpoint

createRunLogWhenSimsFewerThan

Specifies the maximum number of simulation points up to which the details of the best design point are appended to the run log for the *Single Run, Sweeps and Corners* run. ADE XL checks for this limit when the `createRunLogForSweepsCorners` environment variable is set to `ManualTuningOrSimLimited` or `SimLimited`.

In `.cdsenv`:

```
adexl.simulation createRunLogWhenSimsFewerThan int 50
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.simulation" "createRunLogWhenSimsFewerThan" 'int
     50)
```

Valid Values:

A positive integer value between `0` and `1000000`

Default Value: `101`

See also:

- Viewing the Run Log for a Particular Checkpoint

haltCurrentRunAfterPreRunTrigger

Halts the current simulation run after the preRun event occurs. When the preRun event is triggered, you can perform some checks before starting a simulation and set this environment variable to stop the simulation, if required.

For example, if you need to ensure that simulations are not run locally. Instead, they should run on remote computers only, you can use this environment variable, as shown in the code below, to halt the simulation if the distribution method is set to `Local`.

```
; define a callback function in .cdsinit
(define (RunStopper sessionName sdbHandle modeName testName)
(when ((axlGetAttachedJobPolicy)->distributionmethod == "Local")
(printf "Local distribution method used; terminating simulation\n")
(envSetVal "adexl.simulation" "haltCurrentRunAfterPreRunTrigger" 'boolean t))
)

; Connect the callback with the event
(define (connect_handlers session_name)
(axlSessionConnect session_name "preRun" 'RunStopper))

; Register the connected callback to connect the triggers on ADE XL session start
(axlSessionRegisterCreationCallback 'connect_handlers)
```

In `.cdsenv`:

```
adexl.simulation haltCurrentRunAfterPreRunTrigger boolean t
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.simulation" "haltCurrentRunAfterPreRunTrigger"
     'boolean t)
```

Valid Values:

`t`	Halts the current simulation run after the preRun trigger
`nil`	Continues with the current simulation run

Default Value: `nil`

For more information and examples on triggers, refer to Working with ADE (G)XL Signals or Triggers.

ignoreAnalysisCheck

Specifies that ADE XL need not check for existence of analyses before running a simulation. By default, before running a simulation, the tool runs a check to ensure that at least one analysis is defined. However, if the requirement is to run a simulation without any analysis, for example, in case of running a pure digital simulation, you can set this variable to `t` to ignore this check.

In `.cdsenv`:

```
adexl.simulation ignoreAnalysisCheck boolean t
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.simulation" "ignoreAnalysisCheck" 'boolean t)
```

Valid Values:

`t`: Ignores the analysis check.

`nil`: Runs the analysis check to ensure that at least one anlaysis is defined in the setup.

Default Value: `nil`

ignoreDesignChangesDuringRun

Specifies whether ADE XL needs to ignore any design changes in the simulation run that is already running. For more details, refer to Ignoring Design Changes During Run.

In `.cdsenv`:

```
adexl.simulation ignoreDesignChangesDuringRun boolean t
```

In `.cdsinit` or the CIW:

```
envSetVal("adexl.simulation" "ignoreDesignChangesDuringRun" 'boolean
     t)
```

Valid Values:

`t`: Ignores the design changes in the simulation run that is in progress.

`nil`: ADE XL may consider the design changes for the netlist creation and simulation of the pending design points in the current run.

Default Value: `nil`

ignoredLibsForDUT

Specifies the list of libraries that should not be displayed in the *Library* drop-down list in the Design Under Test form. Disabling the display of unnecessary libraries makes it easier to select the correct design under test library for Monte Carlo analysis.

Note: By default, the libraries `analogLib`, `cdsDefTechLib`, and `basic` are not displayed in the Library drop-down list.

In `.cdsenv`:

```
adexl.simulation ignoredLibsForDUT string ""
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.simulation" "ignoredLibsForDUT" 'string "" )
```

Valid Values:

A list of library names separated by spaces.

For example, specify the following in the `.cdsenv` file to ignore the libraries named `lib5` and `lib8`:

```
adexl.simulation ignoredLibsForDUT string "lib5 lib8"
```

includeStatementForNetlistInSimInputFile

Specifies how to include netlist file in the `input.scs` file.

Note: This variable is ignored when the ignoreDesignChangesDuringRun environment variable is set to `t`.

In `.cdsenv`:

```
adexl.simulation includeStatementForNetlistInSimInputFile boolean t
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.simulation"
     "includeStatementForNetlistInSimInputFile" 'boolean nil )
```

Valid Values:

`t`: Includes the netlist file by using the following statement in the `input.scs` file:

```
include "neltist"
```

This helps in saving space consumed by the netlist folder because the netlist is directly included from the netlist file instead of copying the long netlist in the `input.scs` file.

`nil`: Appends the complete netlist to `input.scs` file

Default Values: `nil`

overrideNetlistProcDetection

Controls how messages are displayed when the autoDetectNetlistProcs environment variable is set to `t` and the netlisting mode for incremental simulation runs is set to *Use reference netlist* option in the Reference History form.

In `.cdsenv`:

```
adexl.simulation overrideNetlistProcDetection string ""
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.simulation" "overrideNetlistProcDetection" 'string
     "")
```

Valid Values:

`""`	Displays a message box that indicates that auto-detection and execution of netlist procedures is disabled because the netlisting mode for incremental simulation runs is set to *Use reference netlist*, and prompts you to continue or cancel the incremental simulation run.
`yes`	Displays a warning in the CIW that indicates that auto-detection and execution of netlist procedures is disabled because the netlisting mode for incremental simulation runs is set to *Use reference netlist*, and continues with the incremental simulation run.
`no`	Displays a message box that indicates that auto-detection and execution of netlist procedures is disabled because the netlisting mode for incremental simulation runs is set to *Use reference netlist* option, and requires you to either set the netlisting mode in the Reference History form to *New,* or set the autoDetectNetlistProcs environment variable to `nil`.

Default Value: `""`

overwriteHistory

Controls whether a specified history item is overwritten for subsequent simulation runs.

For more information, see Overwriting a History Item during Subsequent Simulation Runs.

In `.cdsenv`:

```
adexl.simulation overwriteHistory boolean t
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.simulation" "overwriteHistory" 'boolean t)
```

Valid Values:

`t`	Enables overwriting the specified history item for subsequent simulation runs. **Note:** The value of this variable is automatically set to `t` if you specify `1` as the value for the saveLastNHistoryEntries environment variable.
`nil`	Disables overwriting the specified history item for subsequent simulation runs. A new history item will be created for each simulation run.

Default Value: `nil`

See also:

- overwriteHistoryName on page 856
- retainNetlistsOverwriteHistory on page 857

overwriteHistoryName

Specifies the name of the history item to be overwritten for subsequent simulation runs.

For more information, see Overwriting a History Item during Subsequent Simulation Runs.

In `.cdsenv`:

```
adexl.simulation overwriteHistoryName string ""
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.simulation" "overwriteHistoryName" 'string "" )
```

Valid Values:

`Next History Run`	Specifies that the next history item that is created should be overwritten for subsequent simulation runs.
Name of any existing history item. For example, `Interactive.3`	

Default Value: `Next History Run`

See also:

- overwriteHistory on page 855
- retainNetlistsOverwriteHistory on page 857

retainNetlistsOverwriteHistory

Controls whether the netlist information in a history item that is specified to be overwritten is retained for subsequent simulation runs.

For more information, see Overwriting a History Item during Subsequent Simulation Runs.

In `.cdsenv`:

```
adexl.simulation retainNetlistsOverwriteHistory boolean t
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.simulation" "retainNetlistsOverwriteHistory"
     'boolean t)
```

Valid Values:

`t`	Retains the netlist information in the history item for subsequent simulation runs.
`nil`	Deletes the netlist information before each subsequent simulation run.

Default Value: `nil`

See also:

- overwriteHistory on page 855
- overwriteHistoryName on page 856

saveBestNDesignPoints

Specifies the default number of best design points for which to save data when the *Save best* radio button is selected in the *Data Points per Optimization Run* group box on the Save Options form that appears when you choose *Options – Save* in the ADE GXL environment. See saveBestPointsStrategy for more information.

In `.cdsenv`:

```
adexl.simulation saveBestNDesignPoints int 10
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.simulation" "saveBestNDesignPoints" 'int 10 )
```

Valid Values:

Any integer greater than `10`

saveBestPointsStrategy

Specifies which radio button is selected in the *Design Points per Optimization Run* group box on the Save Options form that appears when you choose *Options – Save* in the ADE GXL environment.

In `.cdsenv`:

```
adexl.simulation saveBestPointsStrategy cyclic "Save best"
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.simulation" "saveBestPointsStrategy" 'cyclic "Save
     best" )
```

Valid Values:

`Save all design points`	Saves data for all design points
`Save best`	Saves data for the specified number of best design points; use `saveBestNDesignPoints` to specify the number of points

saveLastNHistoryEntries

Specifies the number of history entries (checkpoints) to save above and beyond any locked entries. See also Specifying How Much Data to Save.

In `.cdsenv`:

```
adexl.simulation saveLastNHistoryEntries int 10
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.simulation" "saveLastNHistoryEntries" 'int 10 )
```

Valid Values:

Any integer greater than `0`

saveNetlistData

Specifies whether to preserve the netlist data generated during a simulation run. This is similar to the *Save Netlists* check box in the Save Options form (See "Specifying Options for Saving Simulation Results" on page 73) that appears when you choose *Options – Save* in the ADE XL environment.

In `.cdsenv`:

```
adexl.simulation saveNetlistData cyclic "Save all points"
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.simulation" "saveNetlistData" 'cyclic "Save all
     points" )
```

Valid Values:

`Save all points`	Preserves netlist data
`Save none`	Deletes netlist data after the simulation run is complete

saveRawData

Specifies whether to preserve the simulation data generated during a simulation run. This is similar to the Save Simulation Data check box in the Save Options form (See “Specifying Options for Saving Simulation Results” on page 73) that appears when you choose *Options – Save* in the ADE XL environment.

In `.cdsenv`:

```
adexl.simulation saveRawData cyclic "Save all points"
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.simulation" "saveRawData" 'cyclic "Save all points"
     )
```

Valid Values:

`Save all points`	Preserves simulation data
`Save none`	Deletes simulation data after the simulation run is complete

showErrorForNonExistingVariables

Checks whether before running a simulation, ADE XL should match the design variables in the Corners Setup form with the list of global variables in the active setup. If the setup for corners uses any design variable that is not present in the active ADE XL setup, simulation is not run and an error is displayed suggesting you to either add that design variable in the active setup or to remove it from the Corners Setup form.

In `.cdsenv`:

```
adexl.simulation showErrorForNonExistingVariables boolean nil
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.simulation" "showErrorForNonExistingVariables"
     'boolean t)
```

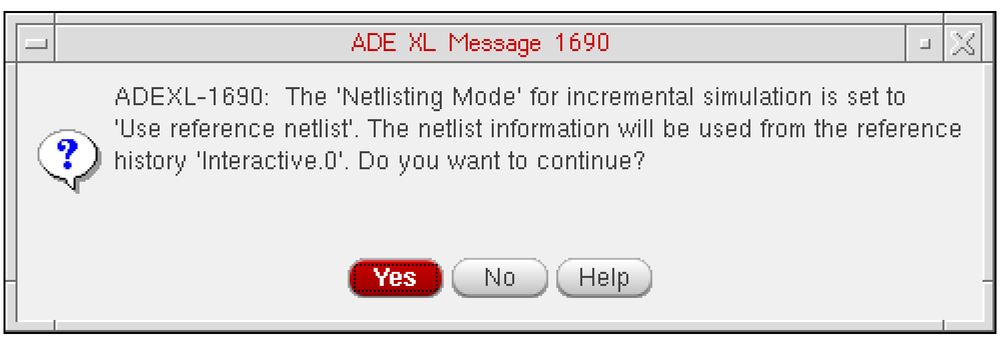
Valid Values:

`t`	Checks for the presence of non-existing variables in the setup for corners.
`nil`	Does not check for the presence of non-existing variables in the setup for corners.

Default Value: `nil`

showWarningForReferenceNetlist

Controls whether the following message box is displayed when you run an incremental simulation with the netlisting mode set to *Use reference netlist* in the Reference History form.



For more information about incremental simulation, see Running an Incremental Simulation on page 497.

In `.cdsenv`:

```
adexl.simulation showWarningForReferenceNetlist boolean t
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.simulation" "showWarningForReferenceNetlist"
     'boolean t)
```

Valid Values:

`t`	Enables the display of the message box when you run an incremental simulation.
`nil`	Disables the display of the message box when you run an incremental simulation.

Default Value: `t`

singleNetlistForAllPoints

By default, ADE XL creates and saves a separate netlist file in the results directory for every design point. For large designs, this results in consuming huge space with same netlist file being saved in multiple directories.

This variable specifies that a common netlist is to be used for all the design points. When this variable is set, a single netlist file is created and a link to that is created in all the point directories. This helps in minimizing the size of the simulation directory.

Note that this variable is ignored in the following cases:

- When the ignoreDesignChangesDuringRun environment variable is set to t
- When device parameterization is enabled

In `.cdsenv`:

```
adexl.simulation singleNetlistForAllPoints boolean t
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.simulation" "singleNetlistForAllPoints" 'boolean t
     )
```

Valid Values:

`t`: Specifies that only a single netlist will be created for all points

`nil`: Creates a separate netlist for each point

Default Value: `nil`

sortVariableValues

By default, while running simulations with corners, ADE XL saves the values of variables and model sections in the order in which they are specified by you. It maintains the same order while saving and displaying the results.

To sort the variable values and model sections in an alphabetical order, set this variable to `t`.

In `.cdsenv`:

```
adexl.simulation sortVariableValues boolean t
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.simulation" "sortVariableValues" 'boolean nil )
```

Valid Values:

`t`: Sorts the variables values and model sections and displays the results in an alphabetical order.

`nil`: Uses the variables values and model sections in the user-specified order.

Default Value: `nil`

warnWhenSimsExceed

Specifies the maximum number of simulations after which the following warning message is displayed:

```
ADEXL-1703: You are about to run more than <max_number> simulations in ADE XL. Do you want to continue ?
```

In `.cdsenv`:

```
adexl.simulation warnWhenSimsExceed int 100
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.simulation" "warnWhenSimsExceed" 'int 100 )
```

Valid Values:

Any number from `0` to `50000`. If `0`, the warning message will not be displayed irrespective of the number of simulations.

Default Value: `100`

adexl.distribute

continueICRPRunOnAbruptGUIExit

Enables continuation and completion of in-process simulations after the ADE XL GUI exits abruptly.

For information about keeping the in-process simulations active, see Continuing the In-Process Simulations After ADE XL GUI Exits on page 475.

In `.cdsenv`:

```
adexl.distribute continueICRPRunOnAbruptGUIExit boolean t
```

In `.cdsinit` or the CIW:

```
envSetVal("adexl.distribute" "continueICRPRunOnAbruptGUIExit"
     'boolean t)
```

Valid Values:

`t`	If the ADE XL GUI exits abruptly, keeps the in-process simulations active. After completion of these simulations, saves their results in the results database.
`nil`	Stops the in-process simulations immediately after the ADE XL GUI exits.

Default Value:

```
nil
```

createUniqueLogsDirForICRPLogs

Specifies if, for each Virtuoso process started from a directory, a unique log subdirectory needs to be created under the `logs_<username>_logs<num>` directory in the Virtuoso working directory. This subdirectory will be used by all the ICRPs started by that Virtuoso process to write their job log files.

By default, this variable is set to `t` and a unique subdirectory is created for each Virtuoso process.

In `.cdsenv`:

```
adexl.distribute  createUniqueLogsDirForICRPLogs boolean t
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.distribute" "createUniqueLogsDirForICRPLogs"
     'boolean t)
```

Valid Values:

`t`	Creates unique subdirectories for each Virtuoso process started from a directory.
`nil`	All Virtuoso processes share a common subdirectory under the `logs_<processID>` directory.

Default Value:

`t`

defaultRunInParallel

Specifies the default option for the Run in field in the Run Options form that appears when you choose *Options – Run Options* in the ADE XL environment.

In `.cdsenv`:

```
adexl.distribute defaultRunInParallel boolean t
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.distribute" "defaultRunInParallel" 'boolean t )
```

Valid Values:

`t`	Sets *Parallel* as the default option for the Run in field in the Run Options form.
`nil`	Sets *Series* as the default option for the Run in field in the Run Options form.

defaultPerRunNumJobs

Specifies a default value for the Specify field in the Run Options form that appears when you choose *Options – Run Options* in the ADE XL environment.

In `.cdsenv`:

```
adexl.distribute defaultPerRunNumJobs int 5
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.distribute" "defaultPerRunNumJobs" 'int 5 )
```

Valid Values:

Any positive integer

generateJobFileOnlyOnError

Specifies if the job log is to be saved only for jobs with an error or for all the jobs. By default, the job log is saved only when a point fails due to an error.

In `.cdsenv`:

```
adexl.distribute generateJobFileOnlyOnError boolean t
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.distribute" "generateJobFileOnlyOnError" 'boolean
     t)
```

Valid Values:

`t`	Saves the job log only for the jobs that fail.
`nil`	Saves the job log for all the jobs.

Default Value: `t`

inferCommandICRPStatusFromProxy

Specifies whether ADE XL should consider the jobs to be interactive or not.

In `.cdsenv`:

```
adexl.distribute inferCommandICRPStatusFromProxy cyclic "Always"
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.distribute" "inferCommandICRPStatusFromProxy"
     'cyclic "Never" )
```

Valid Values:

`Always`	Specifies that the job is always interactive. **Note:** Use this value when you are sure that the job is interactive because if it is not, then the job distribution might not work correctly.
`Never`	Specifies that the job is always interactive.
`GuessFromCommand`	ADE XL will treat the jobs to be interactive only when interactive flags or commands are given. **Note:** ADE XL guesses the known interactive flags or commands only for NC, LSF, and SGE. If you have any other DRMS, ADE XL will not be able to understand whether the jobs are interactive or not. In such case, set this variable to `Always`.

Default Value:

```
GuessFromCommand
```

jobFileDir

Specifies a location where the user logs are saved. By default, the logs are saved in the `logs_<user-name>` directory in the current run directory.

In `.cdsenv`:

```
adexl.distribute jobFileDir string <dir-path>
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.distribute" "JobFileDir" 'string <dir-path> )
```

Valid Values:

Path to the directory where you want to save the user logs.

useAllLingeringJobs

Specifies whether idle or unconfigured jobs must be used when simulations runs are run in series.

For example, assume that you have specified that a maximum of five jobs must be used when simulation runs are run in series. You then run two runs in series, with the first run requiring five jobs to complete and the second run requiring three jobs to complete. When the first run is complete, there will be five idle jobs, but the second run requires only three jobs. If `useAllLingeringJobs` is set to `nil` (the default), the second run will use only three jobs and the remaining two idle jobs will timeout according to the specified linger timeout value. If `useAllLingeringJobs` is set to `t`, the second run will use all the five jobs.

For information about specifying the maximum number of jobs to be used when simulation runs are run in series, see Setting Up Run Options. For information about specifying the linger timeout value for jobs, see Specifying Job Timeouts.

In `.cdsenv`:

```
adexl.distribute useAllLingeringJobs boolean nil
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.distribute" "useAllLingeringJobs" 'boolean nil )
```

Valid Values:

`t`	Uses idle or unconfigured jobs when simulations are run in series.
`nil`	Does not use idle or unconfigured jobs when simulations are run in series.

Default Value:

```
nil
```

maxIPCJobsLimit

Specifies the maximum number of jobs that can be run at any time during your ADE XL session when the distribution method specified in your job policy is *Command*, *Local* or *Remote-Host*. For more information on job policies, see Setting Up Job Policies on page 431.

Tip

If you want to run more than the number of jobs specified using this variable, you must select *LBS* as the distribution method in the Job Policy Setup form.

In `.cdsenv`:

```
adexl.distribute maxIPCJobsLimit int 100
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.distribute" "maxIPCJobsLimit" 'int 100 )
```

Valid Values:

`1` to `115`

Default Value:

`100`

maxJobsIsHardLimit

Controls the interaction between the maximum number of jobs specified for simulation runs, and the number of jobs specified in the *Max Jobs* field on the Job Policy Setup form.

For information about specifying the maximum number of jobs to be used for simulation runs, see Setting Up Run Options on page 451.

In `.cdsenv`:

```
adexl.distribute maxJobsIsHardLimit boolean t
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.distribute" "maxJobsIsHardLimit" 'boolean t )
```

Valid Values:

`t`	Launches only the number of jobs specified in the *Max Jobs* field on the Job Policy Setup form, even if you have specified a greater number of jobs to be used for simulation runs.
`nil`	Launches the maximum number of jobs specified to be used for simulation runs, even if you have specified a lesser number of jobs in the *Max Jobs* field on the Job Policy Setup form.

Default Value:

```
t
```

runTimeoutScaleFactor

Specifies the scale factor to be used to calculate a scaled run timeout value if the `useAsRunTimeout` environment variable is set to `ScaledFromAvgSimTime` or `ScaledFromMaxSimTime`.

In `.cdsenv`:

```
"adexl.distribute" "runTimeoutScaleFactor" int 6
```

In `.cdsinit` or the CIW:

```
envSetVal("adexl.distribute" "runTimeoutScaleFactor" 'int 6)
```

Valid Values:

An integer value between `1` and `1000`

Default Value: `6`

See also:

- `useAsRunTimeout`
- `runTimeoutScalingStartsAfterSimCount`

runTimeoutScalingStartsAfterSimCount

Specifies the maximum number of simulations after which the run timeout value is to be scaled. The scale factor specified by runTimeoutScaleFactor is used to calculate the scaled timeout value.

In .cdsenv:

```
adexl.distribute runTimeoutScalingStartsAfterSimCount int 30
```

In .cdsinit or the CIW:

```
envSetVal("adexl.distribute" "runTimeoutScalingStartsAfterSimCount "
     'int 30)
```

Valid Values:

An integer value between 1 and 1000

Default Value: 20

See also:

- useAsRunTimeout
- runTimeoutScaleFactor

useAsRunTimeout

Specifies the method to be used to calculate the run timeout value for a non-responsive ICRP job. By default, ADE XL uses the run timeout value specified in the job policy. You can use this variable to use an alternate value.

In `.cdsenv`:

```
adexl.distribute useAsRunTimeout cyclic "JobPolicyRunTimeoutValue"
```

In `.cdsinit` or the CIW:

```
envSetVal("adexl.distribute" "useAsRunTimeout" 'cyclic
     "ScaledFromMaxSimTime")
```

Valid Values:

`JobPolicyRunTimeoutValue`	ADE XL uses the run timeout value from the job policy. If that value is set to `NULL`, ADE XL waits for an indefinite time for the ICRP job to confirm that the simulation is complete. In case of a large number of simulations, this can affect the completion of all the pending simulations.
`ScaledFromAvgSimTime`	If the simulation count is less than the limit specified by `runTimeoutScalingStartsAfterSimCount`, ADE XL uses the run timeout value from the job policy. If the simulation count is more than this limit, ADE XL calculates the run timeout value as: `Average sim time * runTimeoutScaleFactor`
`ScaledFromMaxSimTime`	If the simulation count is less than the limit specified by `runTimeoutScalingStartsAfterSimCount`, ADE XL uses the run timeout value from the job policy. If the simulation count is more than this limit, ADE XL calculates the run timeout value as: `Max sim time * runTimeoutScaleFactor`

Default Value: `JobPolicyRunTimeoutValue`

See also:

- runTimeoutScaleFactor
- runTimeoutScalingStartsAfterSimCount

adexl.monte

applySaveOptionsToNetlist

Controls the writing of process and mismatch parameter information in the netlist.

In `.cdsenv`:

```
adexl.monte applySaveOptionsToNetlist boolean t
```

In `.cdsinit`:

```
envSetVal( "adexl.monte" "applySaveOptionsToNetlist" 'boolean t )
```

Valid Values:

`t`	Applies the `saveprocessparams` and `savemismatchparams` options in the netlist depending on the settings for the *Save Process Data* and *Save Mismatch Data* check boxes in the Monte Carlo form. For example, if the *Save Process Data* and *Save Mismatch Data* check boxes are not selected in the Monte Carlo form, the `saveprocessparams` and `savemismatchparams` options are set to `no` in the netlist and Spectre will not write process and mismatch parameter information to the disk. For more information about the `saveprocessparams` and `savemismatchparams` options, see the *Virtuoso Spectre Circuit Simulator Reference*.
`nil`	Writes process and mismatch parameter information in the netlist. When `nil`, the settings for the *Save Process Data* and *Save Mismatch Data* check boxes in the Monte Carlo form are not passed to the netlist.

Default Value: `t`

createStatisticalCornerType

Specifies which method is to be used to create a statistical corner from the Monte Carlo results.

In `.cdsenv`:

```
adexl.monte createStatisticalCornerType cyclic "values"
```

In `.cdsinit`:

```
envSetVal( "adexl.monte" "createStatisticalCornerType" 'cyclic
     "values" )
```

Valid Values:

`sequence`	Create a statistical corner by using a sequence ID of a sample.
`values`	Create a statistical corner by using the statistical parameter values of a sample. **Note:** This requires saving the mismatch data while running Monte Carlo.
`prompt`	Displays the Create Statistical Corner form in which you can confirm which one of the two types mentioned above is to be used to create a statistical corner. The default choice selected in the form is to create the sequence ID-based corner. **Note:** If you have saved the mismatch data while running Monte Carlo, you can choose to create a statistical corner by using the statistical parameter values.
`promptValues`	Displays the Create Statistical Corner form in which you can confirm which one of the two types mentioned above is to be used to create a statistical corner. The default choice selected in the form is to create the statistical parameter-based corner.

Default Value: `sequence`

See also:

- Creating Statistical Corners

incrementalUpdate

Controls the update of Monte Carlo simulation results in the Results tab of the Outputs pane.

In `.cdsenv`:

```
adexl.monte incrementalUpdate boolean t
```

In `.cdsinit`:

```
envSetVal( "adexl.monte" "incrementalUpdate" 'boolean t )
```

Valid Values:

`t`	Monte Carlo simulation results are updated after each iteration of the Monte Carlo run. **Note:** Use the iterationUpdates environment variable to specify the number of iterations of the Monte Carlo run after which the simulation results are updated in the Results tab of the Outputs pane.
`nil`	Monte Carlo simulation results are displayed only after all iterations of the Monte Carlo run are over.

Default Value: `t`

iterationUpdates

Controls the number of iterations of the Monte Carlo run after which simulation results are updated in the Results tab of the Outputs pane.

In `.cdsenv`:

```
adexl.monte iterationUpdates int 10
```

In `.cdsinit`:

```
envSetVal( "adexl.monte" "iterationUpdates" 'int 10 )
```

Valid Values:

Any positive integer

Default Value: 1

maxOutstandingPoints

Specifies the maximum number of samples that are submitted at a time before determining whether to stop a Monte Carlo run. This variable is used when the *Auto Stop Using Significance Test or Auto Stop Using Model Accuracy* option is used.

In `.cdsenv`:

```
adexl.monte maxOutstandingPoints int 100
```

In `.cdsinit`:

```
envSetVal( "adexl.monte" "maxOutstandingPoints" 'int 100 )
```

Valid Values:

Any positive integer

Default Value: 40

Fore more details, see Specifying the maximum outstanding points for Monte Carlo

savedatainseparatedir

Allows saving of raw data (`psf` files) for every Monte Carlo iteration in a separate directory so that you can perform postprocessing operations (like plotting, printing, annotation, re-evaluation, and so on) on individual iterations.

Note: This environment variable is honored only if the *Save Data to Allow Family Plots* check box in the Monte Carlo form is selected.

In `.cdsenv`:

```
adexl.monte savedatainseparatedir boolean t
```

In `.cdsinit`:

```
envSetVal( "adexl.monte" "savedatainseparatedir" 'boolean t )
```

Valid Values:

`t`	Saves raw data (`psf` files) for every Monte Carlo iteration in a separate directory. For example, if there are three iterations, the data for the iterations are saved in directories named `1`, `2` and `3` in the `*libraryName*/*cellName*/adexl/results/data/<*history_item*>` directory
`nil`	Disables the saving of raw data (`psf` files) for every Monte Carlo iteration in a separate directory.

Default Value: `t`

saveProcessOptionDefaultValue

Controls the default setting for the *Save Process Data* check box in the Monte Carlo form.

In `.cdsenv`:

```
adexl.monte saveProcessOptionDefaultValue boolean t
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.monte" "saveProcessOptionDefaultValue" 'boolean t )
```

Valid Values:

`t`	The *Save Process Data* check box in the Monte Carlo form is selected by default (if the settings for this option is not there in the ADE XL setup database).
`nil`	The *Save Process Data* check box in the Monte Carlo form is deselected by default (if the settings for this option is not there in the ADE XL setup database).

Default Value: `t`

saveMismatchOptionDefaultValue

Controls the default setting for the *Save Mismatch Data* check box in the Monte Carlo form. By default, this check box is cleared and ADE XL does not save the mismatch parameters and their values in the associated Monte Carlo results files.

Note: Even if the *Save Mismatch Data* check box is cleared, the mismatch parameters do have an effect on Monte Carlo simulation.

In `.cdsenv`:

```
adexl.monte saveMismatchOptionDefaultValue boolean t
```

In `.cdsinit` (not in CIW):

```
envSetVal( "adexl.monte" "saveMismatchOptionDefaultValue" 'boolean t)
```

Valid Values:

`t`	The *Save Mismatch Data* check box in the Monte Carlo form is selected by default (if the setting for this option is not there in the ADE XL setup database).
`nil`	Clears the *Save Mismatch Data* check box in the Monte Carlo form (if the setting for this option is not there in the ADE XL setup database).

Default Value: `nil`

warnWhenSimsExceed

Specifies a threshold limit for the number of simulations to be run for Monte Carlo. When the number of simulations to be run for Monte Carlo exceeds the specified limit, ADE XL shows the following warning message to confirm if it should continue further:

```
ADEXL-1703: You are about to run more than <max_number> Monte Carlo simulations in
ADE XL. Do you want to continue ?
```

By default, the variable is set to `0` and no check is performed for the maximum number of simulations to be run.

Note: ADE XL does not apply this check when you use the auto stop feature to stop Monte Carlo run based on a specific criteria.

In `.cdsenv`:

```
adexl.monte warnWhenSimsExceed int 100
```

In `.cdsinit` or in CIW:

```
envSetVal( "adexl.monte" "warnWhenSimsExceed" 'int 800)
```

Valid Values:

Any positive integer value

Default Value: `0`

adexl.icrpStartup

binaryName

Specifies the name of the binary to run on the remote host.

In `.cdsenv`:

```
adexl.icrpStartup binaryName string "virtuoso"
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.icrpStartup" "binaryName" 'string "virtuoso" )
```

Valid Values:

Any binary that is valid on the remote host (such as `virtuoso`)

defaultJobPolicy

Specifies the name of the job policy to be used if no job policy is specified in the Job Policy Setup form.

Note the following:

- If no job policy is specified in the Job Policy Setup form or using this variable, the program uses the default job policy settings.
- The job policy settings are overlaid in the following order. A setting from a previous policy is preserved in the final result if not overridden by a subsequent policy.

a. The default job policy settings.

b. The settings in the job policy specified using this variable.

c. The settings in the job policy specified in the Job Policy Setup form.

For more information, see the following examples:

Example 1

If the job policy specified using this variable has a *Max. Jobs* value of `5` and the job policy specified in the Job Policy Setup form has a *Max. Jobs* value of `10`, ADE XL uses a *Max. Jobs* value of `10` for simulation runs.

Example 2

If the job policy specified using this variable has a *Simulation Run Timeout* value of `600` and the job policy specified in the Job Policy Setup form does not have a *Simulation Run Timeout* value, ADE XL uses a *Simulation Run Timeout* value of `600` for simulation runs.

In `.cdsenv`:

```
adexl.icrpStartup defaultJobPolicy string ""
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.icrpStartup" "defaultJobPolicy" 'string "" )
```

Valid Values:

Any valid job policy name

Note the following:

- Do not use the `.jp` job policy file extension when specifying the policy name. For example, specify `myPolicy` instead of `myPolicy.jp`.
- If the job policy name you specify is not defined, setting this environment variable does nothing and the program reverts to the default job policy settings or whatever you select on the Job Policy Setup form in the environment.

enableOutdir

Enables or disables the `-outdir` option, which refers to compiled verilogA module, in the APS or Spectre run script. By default, `-outdir` is included in the script.

In `.cdsenv`:

```
adexl.icrpStartup enableOutdir boolean t
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.icrpStartup" "enableOutdir" 'boolean nil)
```

Valid Values:

`t` : Includes the `-outdir` option, which refers to compiled verilogA module, in the APS or Spectre run script.

`nil` : Removes the `-outdir` option, which refers to compiled verilogA module, from the APS or Spectre run script.

Default Value: `t`

refreshCDF

Specifies when to refresh CDF to consider the base-level CDF values for netlist generation and to ignore the user-level CDF changes.

In `.cdsenv`:

```
adexl.icrpStartup refreshCDF cyclic "Always"
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.icrpStartup" "refreshCDF" 'cyclic "UnlessUserCDF")
```

Valid Values:

`Always` : Always refreshes the CDF to consider the base values.

`Never` : Never refreshes the CDF. Set this value to use user-level CDF settings for ADE XL netlisting.

`UnlessUserCDF` : Refreshes the CDF only if user-level CDF is not available.

Default Value: `Always`

showJobStdout

Specifies whether you want standard output messages from the job submit command (those that the program writes to standard output) to appear in the output area of the Command Interpreter Window (CIW).

Note: You can use this setting to debug problems that might occur while running jobs in Local, Remote-Host or Command mode.

In `.cdsenv`:

```
adexl.icrpStartup showJobStdout boolean nil
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.icrpStartup" "showJobStdout" 'boolean nil )
```

Valid Values:

`t`	Write standard output messages from the job submit command to the CIW
`nil`	Do not write standard output messages from the job submit command to the CIW

showJobStderr

Specifies whether you want standard error messages from the job submit command (those that the program writes to standard error) to appear in the output area of the Command Interpreter Window (CIW).

Note: You can use this setting to debug problems that might occur while running jobs in Local, Remote-Host or Command mode.

In `.cdsenv`:

```
adexl.icrpStartup showJobStderr boolean nil
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.icrpStartup" "showJobStderr" 'boolean nil )
```

Valid Values:

`t`	Write standard error messages from the job submit command to the CIW
`nil`	Do not write standard error messages from the job submit command to the CIW

showOutputLogOnError

Important

This variable is obsolete from the IC6.1.2 release and will be removed in a future release. Instead of specifying this variable, do one of the following:

- Select the *Show output log on error* check box in the Job Policy Setup form.
- Use the `axlSetJobPolicyProperty` SKILL function to specify the default behavior.

  For example, use the following function to display the simulation log file when the program encounters a simulation error:

  ```
  axlSetJobPolicyProperty (<policy_name> "showoutputlogerror" "1")
  ```

  Where the boolean value `"1"` specifies that the simulation log file must be displayed when the program encounters a simulation error. Use the value `"0"` to specify that the simulation log file must not be displayed.

For more information about the `axlSetJobPolicyProperty` SKILL function, see the *Virtuoso Analog Design Environment XL SKILL Reference*.

Specifies whether you want the program to display the simulation log file when it encounters a simulation error. Equivalent to selecting (`t`) or deselecting (`nil`) the *Show output log on error* check box in the Job Policy Setup form).

In `.cdsenv`:

```
adexl.icrpStartup showOutputLogOnError boolean nil
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.icrpStartup" "showOutputLogOnError" 'boolean nil )
```

Valid Values:

`t`	Display the simulation log file when there is a simulation error
`nil`	Do not display the simulation log file

startMaxJobsImmediately

Important

This variable is obsolete from the IC6.1.2 release and will be removed in a future release. Instead of specifying this variable, do one of the following:

- Select the *Start Immediately* check box in the Job Policy Setup form.
- Use the `axlSetJobPolicyProperty` SKILL function to specify the default behavior.

  For example, use the following function to submit the maximum number of simulation jobs immediately when you run a simulation:

  ```
  axlSetJobPolicyProperty (<policy_name> "startmaxjobsimmed" "1")
  ```

  Where the boolean value `"1"` specifies that all specified jobs must be submitted immediately. Use the value `"0"` to submit a job only after a previously submitted job is initialized.

  For more information about the `axlSetJobPolicyProperty` SKILL function, see the *Virtuoso Analog Design Environment XL SKILL Reference*.

Specifies whether you want the program to submit the maximum number of simulation jobs (*Max. Jobs*), or one job for every test, whichever is less, when you start ADE XL.

In `.cdsenv`:

```
adexl.icrpStartup startMaxJobsImmediately boolean t
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.icrpStartup" "startMaxJobsImmediately" 'boolean t )
```

Valid Values:

`t`	Start the maximum number of jobs immediately when you click the *Run Simulation* button on the Run toolbar.
`nil`	Do not start the maximum number of jobs immediately when you click the *Run Simulation* button on the Run toolbar.

adexl.results

defaultBackAnnotationOption

Specifes the default option to be used while backannotating the values from the ADE XL results to the design schematic and ADE XL setup. For details, refer to Backannotating from ADE XL Results on page 583.

In `.cdsenv`:

```
adexl.results defaultBackAnnotationOption cyclic "All variables and
     parameters"
```

In `.cdsinit` or the CIW:

```
envSetVal("adexl.results" "defaultBackAnnotationOption" 'cyclic "All
     variables and parameters")
```

Valid Values:

`"All variables and parameters"`	Backannotates all the global variables and device parameters
`"Only design variables"`	Backannotates only the global variables
`"Only device parameters"`	Backannotates only the device parameters

`"None"`	Does not backannotate any value

Default Value: `"All variables and parameters"`

defaultResultsViewForSweepsCorners

Specifes the default results view for the Single Run, Sweeps, and Corners run mode.

In `.cdsenv`:

```
adexl.results defaultResultsViewForSweepsCorners cyclic "Detail"
```

In `.cdsinit` or the CIW:

```
envSetVal("adexl.results" "defaultResultsViewForSweepsCorners"
     'cyclic "Detail-Transpose")
```

Valid Values:

```
Detail, Detail-Transpose, Optimization, Status, Summary, Yield
```

Default Value: `Detail`

exportPreserveScalingFactors

By default, results are exported to CSV files in the scientific notation format. Set this environment variable to export results in the same format as they are displayed in the Results tab to the CSV file. For more information, see Exporting Results to a HTML or CSV File on page 691.

In `.cdsenv`:

```
adexl.results exportPreserveScalingFactors boolean t
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.results" "exportPreserveScalingFactors" 'boolean t
     )
```

Valid Values:

`t`	Export results as they are displayed in the Results tab to the CSV file.

	`nil`	Export results to CSV files in the scientific notation format.
Default Value:	`nil`	

retainReferenceSimResults

Controls whether the simulation results of the history item on which you ran *Re-run Unfinished/Error Points* are retained in that history item or not. For more information about running *Re-run Unfinished/Error Points*, see Simulating Only Error or Incomplete Points on page 517.

In `.cdsenv`:

```
adexl.results retainReferenceSimResults boolean t
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.gui" "retainReferenceSimResults" 'boolean t )
```

Valid Values:

	`t`	Retains the simulation results of the history item on which you ran *Re-run Unfinished/Error Points*.
	`nil`	Does not retain the simulation results of the history item on which you ran *Re-run Unfinished/Error Points*. As a result, you will not be able to perform postprocessing operations (like plotting, printing, annotation, re-evaluation, and so on) on the history item.
Default Value:	`nil`	

saveDir

Specifies where you want the program to write results database information and run log files for an ADE XL session. When you set this environment variable, the program writes results database information and run log files to *`libraryName`*`/`*`cellName`*`/`*`viewName`*`/results` in the specified `saveDir` location.

If your design library is set up as read-only, you can use this environment variable to specify a writable location. See also the *ADE XL Results Database Location* field on the Save Options form that appears when you choose *Options – Save* in the ADE XL environment.

Note the following:

- If you do not specify a `saveDir`, the program writes results database information and run log files to `*libraryName*/*cellName*/adexl/results/data` in the ADE XL view.
- If you do not specify a `saveDir`, and you open the ADE XL view in read-only mode or do not have write permissions in the ADE XL view, the program writes results database information and run log files to
  `*libraryName*/*cellName*/adexl/results/data/<history_item>`
  in the location specified using the `asimenv.startup projectDir` environment variable. The default setting for this environment variable is `$HOME/simulation`.

In `.cdsenv`:

```
adexl.results saveDir string ""
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.results" "saveDir" 'string "" )
```

Valid Values:

Any valid directory path

saveLocalPsfDir

If the useLocalPsfDir environment variable is set, use this environment variable to specify the path to the local directory on remote systems where the results for distributed simulation jobs run on each remote system must be saved.

Note: Ensure that the specified local directory path exists on all the remote systems on which a distributed simulation is run.

In `.cdsenv`:

```
adexl.results saveLocalPsfDir string ""
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.results" "saveLocalPsfDir" 'string "" )
```

Valid Values:

Any valid directory path

saveResDir

Specifies where you want the program to write simulation results generated during a run. When you set this environment variable, the program writes simulation results to *libraryName*/*cellName*/adexl/results/data/<*history_item*> in the specified `saveResDir` location.

If your design library is set up as read-only, you can use this environment variable to specify a writable location. See also the *Simulation Results Directory Location* field on the Save Options form that appears when you choose *Options – Save* in the ADE XL environment.

Note the following:

- If you do not specify a `saveResDir`, but specify a ADE XL results database location (see saveDir on page 899), the program writes simulation results to *libraryName*/*cellName*/adexl/results/data/<*history_item*> in the ADE XL results database location.
- If you do not specify a `saveResDir` or a ADE XL results database location (see saveDir on page 899), the program writes simulation results to *libraryName*/*cellName*/adexl/results/data/<history_item> in the location specified using the `asimenv.startup projectDir` environment variable. The default setting for this environment variable is `$HOME/simulation`.

In `.cdsenv`:

```
adexl.results saveResDir string ""
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.results" "saveResDir" 'string "" )
```

Valid Values:

Any valid directory path

saveResultsFromHistoryDir

Specifies a default value for the Save Directory field in the Save Results form that appears when you right-click on a history item in the Data View pane, and choose *Save Results*.

In `.cdsenv`:

```
adexl.results saveResultsFromHistoryDir string ""
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.results" "saveResultsFromHistoryDir" 'string "" )
```

Valid Values:

Any valid directory path

evalOutputsOnSimFailure

Controls the evaluation of outputs in case of a failure in analysis. Setup for a particular test can contain more than one analysis. This variable controls how to display outputs for measurements in case the simulation for any particular analysis fails.

When this variable is set to `SkipFailedAnalyses`, the expressions are tied to their corresponding analysis. If the simulation for any analysis fails, the status of outputs related to that analysis is displayed as `sim err`. However, for other outputs that are not dependent on the failing analysis, the result of outputs shows either their result value or `eval error` to correctly display the status.

Note: This environment variable is not supported for the Monte Carlo Sampling run mode.

In `.cdsenv`:

```
adexl.results "evalOutputsOnSimFailure" cyclic "None"
```

In `.cdsinit`

```
envSetVal( "adexl.results" "evalOutputsOnSimFailure" 'cyclic
     "SkipFailedAnalyses" )
```

Note: This value is not honored when specified in the CIW.

Valid Values:

`SkipFailedAnalyses`	Skips the expressions that are tied to failed analysis and calculates the results of other expressions. For the outputs that are tied to the failed analyses, the tool shows `sim err`. You can hover over the cell to display the tooltip with more details on the failed analysis. For the other outputs, which are not tied to a failed analysis, the tool shows the output value in case of successful evaluation. In case of an evaluation error, it displays `eval err`.
`None`	In case of a failed analysis, reports `sim err` for all the expressions.
`All`	All the expressions are evaluated irrespective of whether simulation has passed or failed. If any analysis fails, evaluation is done on partial data that is available in the simulation results directory.

Default Value: `SkipFailedAnalyses`

useLocalPsfDir

By default, the results for distributed simulation runs are saved in the location specified using the `asimenv.startup projectDir` environment variable. Set this environment variable to save the results for distributed simulation jobs run on a remote system in a local directory on that system. Specify the local directory path using the saveLocalPsfDir environment variable.

In `.cdsenv`:

```
adexl.results useLocalPsfDir boolean t
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.results" "useLocalPsfDir" 'boolean t )
```

Valid Values:

	`t`	Saves the results for distributed simulation jobs run on a remote system in a local directory on that system.
	`nil`	Saves the results for distributed simulation runs are saved in the location specified using the `asimenv.startup projectDir` environment variable.
Default Value:	`nil`	

adexl.gui

autoCornerUpdate

Specifies if any changes related to corners or tests in the setup database should be automatically reflected in the *Corners Setup* form that is already open. When set to `t`, the already open *Corners Setup* form is automatically updated to show the changes.

When this variable is set to `nil`, the details are not automatically updated in the *Corners Setup* form. You need to close and re-open the form to view the updated details.

In `.cdsenv`:

```
adexl.gui autoCornerUpdate boolean t
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.gui" "autoCornerUpdate" 'boolean t )
```

Valid Values:

`t`	Automatically updates the *Corners Setup* form with the changes in corner and test details.
`nil`	Does not automatically update the *Corners Setup* form with the changes in corner and test details. You need to close and re-open the form to view the updated details.

Default Value: `nil`

copyMeasurementScripts

Controls whether the OCEAN script file specified for an output of type *OCEAN script* in the Outputs Setup tab is copied to the ADE XL view or used from the original location.

In `.cdsenv`:

```
adexl.gui copyMeasurementScripts boolean t
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.gui" "copyMeasurementScripts" 'boolean t )
```

Valid Values:

`t`	Copies the OCEAN script file to the ADE XL view. Only the file in the ADE XL view is used for simulation runs. As a result, any changes in the original file will not be applied for subsequent simulation runs.
`nil`	Does not copy the OCEAN script file to the ADE XL view. The orginal file is used for simulation runs. As a result, any changes in the original file will be applied for subsequent simulation runs.

Default Value: `t`

Note: Any change in the value of this environment variable will be applied only to new outputs of type *OCEAN script* that you add in the Outputs Setup tab. For example, if you add an OCEAN script output named `OCEAN1` when the value of this variable is `t`, the script file specified for the output is copied to the ADE XL view. However, if you later change the value of this environment variable to `nil`, the script file in the ADE XL view for the OCEAN script output named `OCEAN1` will continue to be used. The original file will not be used.

copyPreRunScripts

Controls whether the pre-run script file specified for a test is copied to the ADE XL view or used from the original location.

In `.cdsenv`:

```
adexl.gui copyPreRunScripts boolean t
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.gui" "copyPreRunScripts" 'boolean t )
```

Valid Values:

`t`	Copies the pre-run script file to the ADE XL view. Only the file in the ADE XL view is used for simulation runs. As a result, any changes in the original file will not be applied for subsequent simulation runs.
`nil`	Does not copy the pre-run script file to the ADE XL view. The original file is used for simulation runs. As a result, any changes in the original file will be applied for subsequent simulation runs.

Default Value: `t`

Note: Any change in the value of this environment variable will be applied only to the new simulation runs. For example, if you add a pre-run script when the value of this variable is `t`, the specified file is copied to the ADE XL view. However, if you later change the value of this environment variable to `nil`, the script file that has already been copied to the ADE XL view will continue to be used. The original file will not be used.

defaultCorners

Specifies the default corners setup file you want the program to load onto the Corners Setup form.

Note: The default corners will be loaded only if no other corners are defined in the Corners Setup form.

In `.cdsenv`:

```
adexl.gui defaultCorners string ""
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.gui" "defaultCorners" 'string "" )
```

Valid Values:

String containing the path to a valid ADE XL corners setup file.

defaultCornerExportFileFormat

Specifies the default format in which the corner details exported from the Corners Setup form are saved.

In `.cdsenv`:

```
adexl.gui defaultCornerExportFileFormat cyclic "CSV"
```

In `.cdsinit` or the CIW:

```
envSetVal("adexl.gui" "defaultCornerExportFileFormat" 'cyclic "SDB")
```

Valid Values:

`CSV`	Specifies that by default the corner details are to be saved in a `.csv` file
`SDB`	Specifies that by default the corner details are to be saved in a `.sdb` file
Default Value	`CSV`

Note: This variable is considered only before opening the Corners Setup form for the first time in an ADE XL session. After the form is opened in a session, the format used or specified in the form is saved as a user preference.

defaultCornerImportFileFormat

Specifies the default format from which the corner details are to be imported into the Corners Setup form.

In `.cdsenv`:

```
adexl.gui defaultCornerImportFileFormat cyclic "CSV"
```

In `.cdsinit` or the CIW:

```
envSetVal("adexl.gui" "defaultCornerImportFileFormat" 'cyclic "SDB")
```

Valid Values:

`CSV`	Specifies that by default the corner details are to be imported from a `.csv` file
`SDB`	Specifies that by default the corner details are to be imported from a `.sdb` file
`PCF`	Specifies that by default the corner details are to be imported from a `.pcf` file

Default Value	`CSV`

Note: This variable is considered only before opening the Corners Setup form for the first time in an ADE XL session. After the form is opened in a session, the format used or specified in the form is saved as a user preference.

disableConstraintsRead

Controls if ADE XL needs to elaborate the Constraint Manager hierarchy to find the matched parameter constraints and import them to the ADE XL setup. By default, this variable is set to `nil` and ADE XL elaborates the Constraint Manager hierarchy. Set this variable to `t` in any one of the following two cases:

- When the Constraints Manager does not contain matched parameters for your design.
- When you do not wish to automatically import all the matched parameters from the Constraints Manager to ADE XL.

Disabling constraints read helps in improving the performance of ADE XL.

In `.cdsenv`:

```
adexl.gui disableConstraintsRead boolean t
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.gui" "disableConstraintsRead" 'boolean t )
```

Valid Values:

`t`	Disables elaboration of the Constraint Manager hierarchy to find matched parameter constraints.
`nil`	Enables elaboration of the Constraint Manager hierarchy to find matched parameter constraints.

Default Value: `nil`

disableNominalSimulation

Controls whether the *Nominal Corner* check box in the Run Summary assistant pane is selected or deselected by default when you start ADE XL. If the *Nominal Corner* check box is selected, the simulator runs nominal corner simulation.

In `.cdsenv`:

```
adexl.gui disableNominalSimulation boolean t
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.gui" "disableNominalSimulation" 'boolean t )
```

Valid Values:

`t`	Deselects the *Nominal Corner* check box by default when you start ADE XL.
`nil`	Selects the *Nominal Corner* check box by default when you start ADE XL.

Default Value: `nil`

disableRunInReadOnly

Controls whether simulations can be run when the ADE XL view is opened in read-only mode. For more information, see Working with Read-Only ADE XL Views.

In `.cdsenv`:

```
adexl.gui disableRunInReadOnly boolean t
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.gui" "disableRunInReadOnly" 'boolean t )
```

Valid Values:

`t`	Does not allows simulations to be run in ADE XL when the ADE XL view is opened in read-only mode.
`nil`	Allows simulations to be run in ADE XL when the ADE XL view is opened in read-only mode.

Default Value: `nil`

disableSimulationsDefault

Specifies whether the nominal corner or other corners (corners other than the nominal corner) are enabled or disabled by default when you create a new ADE XL view.

Tip

This environment variable can be used along with the defaultCorners environment variable to only run the provided list of corners.

In `.cdsenv`:

```
adexl.gui disableSimulationsDefault cyclic "nominal"
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.gui" "disableSimulationsDefault" 'cyclic "nominal"
    )
```

Valid Values:

`nominal`	Disables the nominal corner when you create a new ADE XL view. The *Nominal Corner* check box in the Run Summary assistant pane is deselected by default. **Note:** If tests are enabled in the ADE XL view but no corners are specified, the *Nominal Corner* check box in the Run Summary assistant pane is automatically enabled so that simulations can be run.
`corners`	Disables other corners (corners other than the nominal corner) when you create a new ADE XL view. The *Corner* check box in the Run Summary assistant pane is deselected by default.
`none`	Enables all the corners when you create a new ADE XL view.

Default Value:

`none`

filterCDFParamsWithZeroOrNegat iveOneDefValue

Displays or hides CDF parameters in the Variables and Parameters assistant that have have default value set as `0` or `-1`. By default, variables that have default values set to any one of t, "", "0", "-1", 0, or -1 are not displayed in the Variables and Parameters assistant.

In `.cdsenv`:

```
adexl.gui filterCDFParamsWithZeroOrNegativeOneDefValue boolean t
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.gui"
     "filterCDFParamsWithZeroOrNegativeOneDefValue" boolean nil )
```

Valid Values:

`t`	Hides or filters out those CDF parameters from the Variables and Parameters assistant that have have default value set as `0` or `-1`.
`nil`	Displays those CDF parameters in the Variables and Parameters assistant that have have default value set as `0` or `-1`.

Default Value:

```
t
```

LimitModelSections

Specifies how to handle errors when the model section name specified for a corner is not found in the corresponding model file or PCF file.

In `.cdsenv`:

```
adexl.gui LimitModelSections cyclic "InModelFile"
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.gui" "LimitModelSections" 'cyclic "LimitedList" )
```

Valid Values:

`InModelFile`	If the section name specified for a corner is not present in model file, an error is displayed in Corners Setup form.
`LimitedList`	If the specified section name is not present in the PCF file, an error is displayed in Corners Setup form.
`No`	If the specified section name is not present in the model file or the PCF file, no error is displayed in Corners Setup form. However, an error is displayed during the simulation run.

Default Value:

`No`

maxNotesLength

Specifies the maximum character limit for a note that can be added to an ADE XL element.

In `.cdsenv`:

```
adexl.gui maxNotesLength int 400
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.gui" "maxNotesLength" 'int 400 )
```

Valid Values:

Any positive integer between `1` and `5120`

Default Value:

`512`

See also:

- Adding Notes to a Test.

maxNotesRowsDisplay

Specifies the maximum limit of the number of lines of a note to be displayed in the tooltip for an ADE XL element.

In `.cdsenv`:

```
adexl.gui maxNotesRowsDisplay int 4
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.gui" "maxNotesRowsDisplay" 'int 4 )
```

Valid Values:

Any positive integer

Default Value:

`10`

See also:

- Adding Notes to a Test.

mismatchPairs

Specifies the default maximum number of device parameters for which mismatch results are displayed in the Show Mismatch form.

If the maximum number of device parameters in your design is lesser than this number, mismatch results are displayed for all the device parameters in your design. If the maximum number of device parameters in your design is greater than this number, mismatch results are displayed only for the number of device parameters specified using this environment variable.

For example, if you specify the value of the `mismatchPairs` variable as `20` and your design has `10` device parameters, mismatch results are displayed only for `10` device parameters. However, if your design has `25` design parameters, mismatch results are displayed only for the most sensitive `20` device parameters.

In `.cdsenv`:

```
adexl.gui mismatchPairs int 100
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.gui" "mismatchPairs" 'int 100 )
```

Valid Values:

Any positive integer.

Default Value:

`200`

modelSectionFilterFunction

Specifies a function used to filter the list of model sections displayed in the *Section* drop-down list in the Add/Edit Model Files form (see Adding Model Files to a Corner on page 230) that is opened from the Corners Setup form.

In `.cdsenv`:

```
adexl.gui modelSectionFilterFunction string ""
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.gui" "modelSectionFilterFunction" 'string "" )
```

Valid Values:

String containing the name of a defined function that has the signature:

```
(t_fileName l_initialSectionList)
    => l_filteredSectionList
```

For example, if you have a model file named `mymodel.scs` that has the sections `tt`, `ss`, `fs`, and `unused`, do the following if you do not want the section `unused` to be displayed in *Section* drop-down list in the Add/Edit Model Files form that is opened from the Corners Setup form:

1. In your `.cdsinit` file or the CIW, define a function, say `CornerSectionFilt`, that specifies that the section `unused` must be filtered. For example:

```
procedure( CornerSectionFilt(model_file_name input_sections)
  let( ((file_tail car(last(parseString(model_file_name "/"))))
  output_sections)
  if( file_tail == "mymodel.scs" then
    output_sections = setof(name input_sections (name !=
"unused"))
  else
    output_sections = input_sections)
  output_sections))
```

2. Specify the `modelSectionFilterFunction` environment variable. For example, specify the following in your `.cdsenv` file:

```
adexl.gui modelSectionFilterFunction string "CornerSectionFilt"
```

numberOfBestPointsToView

Specifies the maximum number of best design points you want the program to display on the Results tab of the Outputs pane when you run an optimization in the Analog Design Environment GXL environment.

In `.cdsenv`:

```
adexl.gui numberOfBestPointsToView int 10
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.gui" "numberOfBestPointsToView" 'int 10 )
```

Valid Values:

Any positive integer

Default Value: 10

openDesignAccessMode

Specifies the default mode in which designs are opened when you right-click on a test in the Data View pane and choose *Open Design in Tab*.

In `.cdsenv`:

```
adexl.gui openDesignAccessMode cyclic "w"
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.gui" "openDesignAccessMode" 'cyclic "w" )
```

Valid Values:

`r`	Opens designs in read mode.
`a`	Opens designs in append mode.
`w`	Opens designs in write mode.

openSchInWin

Controls whether a schematic opened from the Outputs Setup tab will be displayed in a new window, or in a new tab in the current window.

In `.cdsenv`:

```
adexl.gui openSchInWin boolean t
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.gui" "openSchInWin" 'boolean t )
```

Valid Values:

`t`	Displays the schematic opened from the Outputs Setup tab in a new window. For example, if you right click on a test name in the Outputs Setup tab and choose *To be Plotted*, the schematic is displayed in a new window.
`nil`	Displays the schematic opened from the Outputs Setup tab in a new tab in the current window.

openTerminalCommand

Specifies the shell command you want the program to use when you select *Open Terminal* from the right-click pop-up menu for a history item on the Data assistant pane. The program uses the shell command to open a terminal window in the directory containing results for the selected history item.

In `.cdsenv`:

```
adexl.gui openTerminalCommand string ""
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.gui" "openTerminalCommand" 'string "" )
```

Valid Values:

Any valid command string

The default shell command when `openTerminalCommand` is not set (or when it is set to the empty string as shown above) is

`xterm -T "`*`historyResultsDirectory`*`"`

where *`historyResultsDirectory`* is the name of the directory containing results for the selected history item (such as `Interactive.0` or `GlobalOpt.1` or, for *ImproveYield* history items, the *historychildren* item name, such as `ImproveYield.0.GlobalOpt.0`). The `xterm` command must be in your path.

Note: You must not put `&` at the end of the `openTerminalCommand` string.

pcfPrependBasePath

By default, when you import a process customization file (PCF) or design customization file (DCF) to create corners, the related process model file names are displayed in the Corners Setup form. However, you must specify the path to the directory containing the process models as an include path in the Simulation Files Setup form so that the simulator can read the process model files from the specified directory.

Use this environment variable to control whether the path to the process model files are included in the Corners Setup form so that you need not specify the path to the directory containing the process models as an include path in the Simulation Files Setup form.

For more information about importing PCF or DCF files, see Importing Corners from Customization Files.

In `.cdsenv`:

```
adexl.gui pcfPrependBasePath boolean t
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.gui" "pcfPrependBasePath" 'boolean t )
```

Valid Values:

`t`	Includes process model file paths in the Corners Setup form when you import PCF or DCF files.
`nil`	Does not include process model file paths when you import PCF or DCF files.

Default Value: `nil`

setHistoryPrefixToSetupStateNameOnLoad

Specifies if the name of the loaded setup state should be used as a prefix in the history name. By default, when you load a setup state and run simulation, the history name takes the setup state name as a prefix. To use `Interactive` as a prefix, set this variable to `nil`.

In `.cdsenv`:

```
adexl.gui setHistoryPrefixToSetupStateNameOnLoad boolean t
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.gui" "setHistoryPrefixToSetupStateNameOnLoad"
     'boolean t )
```

Valid Values:

`t`	Displays the setup state name as a prefix in the history name.
`nil`	Displays `interactive` as a prefix in the history name.

Default Value: `t`

setupFormDefaultEnabled

Specifies the check boxes that will be selected by default in the:

- *What to Import* group box in the Import Setup form (see Importing the Simulation Setup)
- *What to Export* group box in the Export Setup form (see Exporting the Simulation Setup)
- *What to Save* group box in the Save Setup State form (see Creating or Updating a Setup State)
- *What to Load* group box in the Load Setup State form (see Loading a Setup State)

In `.cdsenv`:

```
adexl.gui setupFormDefaultEnabled string "all"
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.gui" "setupFormDefaultEnabled" 'string "all" )
```

Valid Values:

`all`	All the check boxes are selected by default.
`""`	Empty string. All the check boxes are deselected by default.

`tests`	List of option names separated by a comma, semicolon or space. The check boxes for the specified options are selected by default.
`vars`	For example, to select the *Tests*, *Variables* and *Parameters* check boxes, specify the value as:
`parameters`	`"tests, vars, parameters"`
`currentmode`	For more information about these options, see the following topics:
`allsweepsenabled`	■ Importing the Simulation Setup on page 778
`allcornersenabled`	■ Exporting the Simulation Setup on page 781
`defaultcornerenabled`	■ Creating or Updating a Setup State on page 785
`runoptions`	■ Loading a Setup State on page 787
`specs`	
`corners`	
`modelgroups`	
`extensions`	

Default Value: `all`

setupFormDefaultLoadOperation

Specifies the default value of the *Operation* drop-down list in the Load Setup State form (see Loading a Setup State on page 787) and the Import Setup form (see Importing the Simulation Setup on page 778).

In `.cdsenv`:

```
adexl.gui setupFormDefaultLoadOperation cyclic "retain"
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.gui" "setupFormDefaultLoadOperation" 'cyclic
     "retain" )
```

Valid Values:

`retain`

`merge`

`overwrite`

For more information about these options, see the following topics:

- Loading a Setup State on page 787
- Importing the Simulation Setup on page 778.

Default Value: `retain`

significantDigits

Specifies the number of significant digits you want the program to display for values in the *Nominal* column on the Results tab.

In `.cdsenv`:

```
adexl.gui significantDigits int 4
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.gui" "significantDigits" 'int 4 )
```

Valid Values:

Any integer from `2` through `15`

specComparisonMode

Specifies the default comparison mode in the Spec Comparison form. For more information, see Comparing Results on page 659.

In `.cdsenv`:

```
adexl.gui specComparisonMode cyclic "Histories"
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.gui" "specComparisonMode" 'cyclic "Histories" )
```

Valid Values:

`Histories`

`Design Points`

Default Value: `Histories`

toolbarButtonStyle

Specifies whether the *Match Parameters* and *Ratio Matched Parameters* buttons on the Parameters tab of the Variables and Parameters assistant pane use an icon or text.

In `.cdsenv`:

```
adexl.gui toolbarButtonStyle cyclic "Histories"
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.gui" "toolbarButtonStyle" 'cyclic "icon" )
```

Valid Values:

`icon`	If `icon`, the *Match Parameters* button uses the  icon and the *Ratio Matched Parameters* button uses the  icon.
`text`	If `text`, the *Match Parameters* button uses the text *Match* and the *Ratio Matched Parameters* button uses the text *Ratio*.

Default Value: `icon`

yieldViewShowDefault

Specifies the default columns that you want to be displayed in the yield view for Monte Carlo results.

In `.cdsenv`:

```
adexl.gui yieldViewShowDefault string "\"Min\" \"Target\" \"Max\"
     \"Mean\" \"Sigma\" \"Cpk\""
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.gui" "yieldViewShowDefault " 'string "\"Min\"
     \"Target\" \"Max\" \"Mean\" \"Sigma\" \"Cpk\"" )
```

Valid Values:

`string`	A string with the list of space-separated column names

Default Value:	`\"Min\" \"Target\" \"Max\" \"Mean\" \"Sigma\" \"Sigma to Target\" \"Cpk\"`

adexl.cpupdtr

copyResultsData

Copies the simulation results when you copy an ADE XL view. For more information, see Copying Everything in an ADE XL View on page 772.

Note: It is recommended to add this environment variable only in the `.cdsenv` file. The variable is not used when specified in the `.cdsinit` file.

In `.cdsenv`:

```
adexl.cpupdtr copyResultsData boolean  t
```

Valid Values:

	`t`	Copy the simulation results data.
	`nil`	Do not copy the simulation results data.
Default Value:	`nil`	

adexl.datasheet

CSSFile

Specifies a custom Cascading Style Sheet (CSS) file for controlling the formatting of the main datasheet file and the datasheet files that contain the results information for each test. For more information, see Customizing the Datasheet Format and Structure on page 830.

In `.cdsenv`:

```
adexl.datasheet CSSFile string " "
```

In the CIW:

```
envSetVal( "adexl.datasheet" "CSSFile" 'string " " )
```

Valid Values:

String containing the path to a custom CSS file for the datasheet pages.

customFiles

Specifies the files to be copied to the datasheet directory when a datasheet is created.

Tip

If you have customized your datasheet format using custom XSLT stylesheets, you can use this environment variable to copy files such as the image file for your company logo and other support files that are required by the custom stylesheets. For more information about customizing the datasheet format, see Customizing the Datasheet Format and Structure on page 830.

In `.cdsenv`:

```
adexl.datasheet customFiles string " "
```

In the CIW:

```
envSetVal( "adexl.datasheet" "customFiles" 'string " " )
```

Valid Values:

String containing the path to a file or directory.

- If the path to a file is specified, only that file is copied to the datasheet directory when a datasheet is created.
- If the path to a directory is specified, all the files and sub-directories in the directory are copied to the datasheet directory when a datasheet is created.

mainDocXSLFile

Specifies a custom XSLT stylesheet for controlling the structure of the main datasheet file. For more information, see Customizing the Datasheet Format and Structure on page 830.

In `.cdsenv`:

```
adexl.datasheet mainDocXSLFile string " "
```

In the CIW:

```
envSetVal( "adexl.datasheet" "mainDocXSLFile" 'string " " )
```

Valid Values:

String containing the path to a custom XSLT file for the main datasheet page.

testDocXSLFile

Specifies a custom XSLT file for controlling the structure of the datasheet files that contain the results information for each test. For more information, see Customizing the Datasheet Format and Structure on page 830.

In `.cdsenv`:

```
adexl.datasheet testDocXSLFile string " "
```

In the CIW:

```
envSetVal( "adexl.datasheet" "testDocXSLFile" 'string " " )
```

Valid Values:

String containing the path to a custom XSLT file for the datasheet pages for each test.

adexl.testEditor

showAllMenus

The ADE XL Test Editor window is a customized version of the Virtuoso Analog Design Environment L (ADE L) session window. By default, only the ADE XL specific menus are displayed in the ADE XL Test Editor window. This environment variable allows you to display all the ADE L menus in the ADE XL Test Editor window so that you can run simulations on individual tests from the ADE XL Test Editor window. For more details, refer to <u>Opening the Test Editor Window</u>.

In `.cdsenv`:

```
adexl.testEditor showAllMenus boolean t
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.testEditor" "showAllMenus" 'boolean t )
```

Valid Values:

`t`	Displays all the ADE L menus in the ADE XL Test Editor window.
`nil`	Displays only the ADE XL specific menus in the ADE XL Test Editor window.

Default Value: `nil`

asimenv.startup

copyDesignVarsFromCellview

Controls copy of design variables from a cellview to the design variables of ADE XL test.

In `.cdsenv`:

```
asimenv.startup copyDesignVarsFromCellview boolean t
```

Valid Values:

`t`	Enables copy of design variables from the cellview property to a test
`nil`	Stops copying design variables from the cellview property.

Default Value: `t`

adexl.oceanxl

includeSimLogInJobLog

Controls whether the simulation log is to be included in the job log generated by the ICRP for an OCEAN run. By default, the log for an OCEAN run does not include the simulator output because for large simulations, this can result in large job log files. To save the simulator log in the ICRP job log for an OCEAN run, set this variable to `t` before running a simulation.

In `.cdsenv`:

```
adexl.oceanxl includeSimLogInJobLog boolean t
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.oceanxl" "includeSimLogInJobLog" 'boolean t )
```

Valid Values:

`t`	Includes the simulation log in the job log for an OCEAN run.
`nil`	Does not include the simulation log in the job log for an OCEAN run.

Default Value: `nil`

adexl.plotting

histogramBins

Specifies the default value for the *Number of Bins* field on the Plot Histogram form.

In `.cdsenv`:

```
adexl.plotting histogramBins int 10
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.plotting" "histogramBins" 'int 10 )
```

Valid Values:

```
A positive integer value
```

Default Value: `10`

histogramType

Sets the default value for the *Type* drop-down list on the Plot Histogram form.

In `.cdsenv`:

```
adexl.plotting histogramType string "pass/fail"
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.plotting" "histogramType" 'string "pass/fail" )
```

Valid Values:

`pass/fail`	Plots a standard histogram with pass/fail spec markers
`standard`	Plots a standard histogram
`cumulative line`	Plots a cumulative line histogram
`cumulative box`	Plots a cumulative box histogram

Default Value: `pass/fail`

histogramQQPlot

Specifies the default value for the *Normal Quantile Plot* annotation option on the Plot Histogram form.

In `.cdsenv`:

```
adexl.plotting histogramQQPlot boolean t
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.plotting" "histogramQQPlot" 'boolean nil )
```

Valid Values:

`t`	Plots the quantile plots (Q-Q plots) along with the histogram
`nil`	Does not plot the quantile plots (Q-Q plots)

Default Value: `nil`

plotScalarExpressions

Controls whether expressions that evaluate to scalar values are automatically plotted after the simulation run is complete.

In `.cdsenv`:

```
adexl.plotting plotScalarExpressions boolean t
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.plotting" "plotScalarExpressions" 'boolean t )
```

Valid Values:

`t`	Enables automatic plotting of expressions that evaluate to scalar values after the simulation run is complete.
`nil`	Disables automatic plotting of expressions that evaluate to scalar values after the simulation run is complete.

Default Value: `t`

plotSignals

Controls whether signals are automatically plotted after the simulation run is complete.

In `.cdsenv`:

```
adexl.plotting plotSignals boolean t
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.plotting" "plotSignals" 'boolean t )
```

Valid Values:

`t`	Enables automatic plotting of signals after the simulation run is complete.
`nil`	Disables automatic plotting of signals after the simulation run is complete.

Default Value: `t`

plotType

Specifies the default plotting type to be used for all tests. For more information, see Setting Default Plotting Options for All Tests.

In `.cdsenv`:

```
adexl.plotting plotType cyclic "Auto"
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.plotting" "plotType" 'cyclic "Auto" )
```

Valid Values:

`Auto`	Automatically plots outputs after the simulation run is complete. For every subsequent simulation run, a new graph replaces the existing graph.
`Refresh`	Automatically plots outputs after the simulation run is complete, but refreshes the existing graph in the same window.
`None`	Disables automatic plotting of results after the simulation run.

Default Value: `Auto`

plotWaveExpressions

Controls whether expressions that evaluate to waveforms are automatically plotted after the simulation run is complete.

In `.cdsenv`:

```
adexl.plotting plotWaveExpressions boolean t
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.plotting" "plotWaveExpressions" 'boolean t )
```

Valid Values:

`t`	Enables automatic plotting of expressions that evaluate to waveforms after the simulation run is complete.

`nil`	Disables automatic plotting of expressions that evaluate to waveforms after the simulation run is complete.

Default Value: `t`

showHistogramDensity

Specifies the default value for the *Density Estimator* annotation option on the Plot Histogram form.

In `.cdsenv`:

```
adexl.plotting showHistogramDensity boolean t
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.plotting" "showHistogramDensity" 'boolean nil )
```

Valid Values:

`t`	Plots a curve on the histogram that estimates the distribution concentration.
`nil`	Does not plot the curve.

Default Value: `t`

showHistogramDeviation

Specifies the default value for the *Std Dev Lines* annotation option on the Plot Histogram form.

In `.cdsenv`:

```
adexl.plotting showHistogramDeviation boolean t
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.plotting" "showHistogramDeviation" 'boolean nil )
```

Valid Values:

`t`	Shows the standard deviation lines in the graph indicating the mean, mean – standard deviation, and mean + standard deviation values.

	`nil`	Does not show the standard deviation lines with the histogram.
Default Value:	`t`	

showHistogramPoints

Controls the display of data points on histograms to enable cross-selection between ADE XL results table and the histogram plotted in the Virtuoso Visualization and Analysis XL window.

By default, the histogram data points are visible and the bars are filled with a transparent or alpha color to make the points clearly visible. You can select a histogram point to cross-select the corresponding result in the ADE XL Results tab. Set this variable to `nil` to disable cross-selection from histogram points. In this case, the bars are filled with solid color.

Note: You can also show or hide the data points on histograms using the Trace Properties form or by selecting or clearing the *Symbols On* command in the context-sensitive menu of histograms.

In `.cdsenv`:

```
adexl.plotting showHistogramPoints boolean t
```

In `.cdsinit` or the CIW:

```
envSetVal( "adexl.plotting" "showHistogramPoints" 'boolean nil )
```

Valid Values:

	`t`	Shows data points on histograms. The histogram bars are filled with a transparent or alpha color to make the data points clearly visible. You can change the style of data points by using the *Symbols* command on the context-sensitive menu of histogram.
	`nil`	Hides data points on histograms. Histogram bars are filled with solid colors and points are not visible.
Default Value:	`t`	

Also see: Cross-Probing ADE XL Results from Histogram Plots

asimenv.plotting

specMarkers

Controls whether spec markers should be displayed in the graphs plotted after simulation.

In `.cdsenv`:

```
asimenv.plotting specMarkers boolean nil
```

In `.cdsinit` or the CIW:

```
envSetVal( "asimenv.plotting" "specMarkers" 'boolean nil )
```

Valid Values:

`t`	Displays spec markers in the graphs. When this variable is set to `t`, the *Spec Markers* graph annotations option on the the ADE XL Printing/ Plotting Options form is enabled.
`nil`	Does not display spec markers in the graphs.

Default Value: `nil`

Index

Symbols

A

B

C

F

G

H

I

J

P

T

V