cadence™

# Virtuoso ADE Explorer User Guide

**Product Version IC23.1**
**November 2023**

© 2023 Cadence Design Systems, Inc. All rights reserved.
Cadence Design Systems, Inc. (Cadence), 2655 Seely Ave., San Jose, CA 95134, USA.

**Trademarks**: Trademarks and service marks of Cadence Design Systems, Inc. contained in this document are attributed to Cadence with the appropriate symbol. For queries regarding Cadence's trademarks, contact the corporate legal department at the address shown above or call 800.862.4522.

All other trademarks are the property of their respective holders.

**Restricted Permission:** This publication is protected by copyright law and international treaties and contains trade secrets and proprietary information owned by Cadence. Unauthorized reproduction or distribution of this publication, or any portion of it, may result in civil and criminal penalties. Except as specified in this permission statement, this publication may not be copied, reproduced, modified, published, uploaded, posted, transmitted, or distributed in any way, without prior written permission from Cadence. Unless otherwise agreed to by Cadence in writing, this statement grants Cadence customers permission to print one (1) hard copy of this publication subject to the following conditions:

1. The publication may be used only in accordance with a written agreement between Cadence and its customer.
2. The publication may not be modified in any way.
3. Any authorized copy of the publication or portion thereof must include all original copyright, trademark, and other proprietary notices and this permission statement.
4. The information contained in this document cannot be used in the development of like products or software, whether for internal or external use, and shall not be used for the benefit of any other party, whether or not for consideration.

**Disclaimer:** Information in this publication is subject to change without notice and does not represent a commitment on the part of Cadence. Except as may be explicitly set forth in such agreement, Cadence does not make, and expressly disclaims, any representations or warranties as to the completeness, accuracy or usefulness of the information contained in this document. Cadence does not warrant that use of such information will not infringe any third party rights, nor does Cadence assume any liability for damages or costs of any kind that may result from use of such information.

Cadence is committed to using respectful language in our code and communications. We are also active in the removal and replacement of inappropriate language from existing content. This product documentation may however contain material that is no longer considered appropriate but still reflects long-standing industry terminology. Such content will be addressed at a time when the related software can be updated without end-user impact.

**Restricted Rights:** Use, duplication, or disclosure by the Government is subject to restrictions as set forth in FAR52.227-14 and DFAR252.227-7013 et seq. or its successor

# Contents

1
Getting Started with ADE Explorer . . . . . 35
Introducing Virtuoso ADE Explorer . . . . . 35
Licensing Requirements . . . . . 36
Opening Virtuoso ADE Explorer . . . . . 37
Opening ADE Explorer From CIW . . . . . 37
Opening ADE Explorer From Virtuoso Schematic Editor . . . . . 42
Opening ADE Explorer Using Schematic Editor Workspace . . . . . 48
ADE Explorer Toolbars in Virtuoso Schematic Editor . . . . . 49
ADE Explorer Menu in Schematic . . . . . 50
Virtuoso ADE Explorer Graphical User Interface . . . . . 51
Menu Bar . . . . . 52
Toolbars . . . . . 68
Assistants . . . . . 73
Outputs Setup Pane and Tabs . . . . . 85
Results Tab . . . . . 86
Run Preview Tab . . . . . 87
Status Bar . . . . . 88
Switching Between ADE Explorer and ADE Assembler . . . . . 89
. . . . . 92

2
Setting Up Explorer Environment . . . . . 93
Prerequisites For Running Explorer . . . . . 95
Choosing a Design . . . . . 95
Choosing a Simulator . . . . . 96
Setting the Simulation Temperature . . . . . 97
Choosing an Analysis . . . . . 98
Choosing Model Library Path . . . . . 99
Specifying Simulator Options . . . . . 102
Spectre . . . . . 102


© 2023 Cadence Design Systems, Inc.

Simulator Options Form . . . 104
Main . . . 104
Algorithm . . . 107
Component . . . 110
Check . . . 115
Annotation . . . 118
Miscellaneous . . . 121
Spectre X . . . 130
Spectre FX . . . 136
Virtuoso Accelerated Parallel Simulator (APS) . . . 138
Virtuoso UltraSim Simulator Interface . . . 139
Virtuoso AMS Simulator Interface . . . 139
Virtuoso HSPICE Simulator Interface . . . 139
Setting up High-Performance Simulation Options . . . 140
Virtuoso Accelerated Parallel Simulator (APS) . . . 154
Setting Up Simulation Files . . . 155
Specifying Simulation Files . . . 157
Specifying Simulation Files Using Variables . . . 160
Using Shell Environment Variables to Specify the Paths . . . 161
Enabling and Disabling Simulation Files . . . 162
Editing Simulation Files . . . 162
Deleting Simulation Files . . . 163
Setting up Simulator Environment Options . . . 164
Setting up Design Variables . . . 169
Adding a New Variable . . . 169
Editing Values . . . 175
Deleting Values . . . 176
Copying Values From and To Schematic . . . 176
Displaying Values on the Schematic . . . 177
Finding Design Variables in Schematic . . . 177
Performing Parameterization in Design Variables . . . 177
Adding Notes . . . 180
Creating and Displaying Netlists . . . 181
The .simrc File . . . 181
Incremental Netlisting . . . 182
Creating and Displaying a Netlist . . . 182


© 2023 Cadence Design Systems, Inc.

Variables for Customizing Netlist Generation . . . 185
Setting up Stimuli . . . 189
Setting up Stimuli Using the Stimuli Assignment Form . . . 189
Creating and Managing Stimuli . . . 191
Specifying a Stimulus File . . . 210
Example of a Spectre Stimulus File . . . 210
Setting up Cosimulation Options . . . 211
Setting up MDL Mode . . . 213
Saving the Current Setup . . . 215
Importing a Saved Setup . . . 218
Specifying Simulation Results Directory Location . . . 225
How does ADE Explorer save the results in the results database? . . . 226
Saving Results in ADE Explorer . . . 229
Saving an OCEAN Script . . . 234
Closing a Session . . . 236
Configuring and Customizing ADE Explorer Settings . . . 237
Customizing Your .cdsinit File . . . 237
Customizing Your .cdsenv File . . . 237
Customizing Your Menus File . . . 237
Setting UNIX Environment Variables . . . 238
RF Options . . . 239
Using Signals or Triggers . . . 242
Reserved Words . . . 243

3
Setting Up Analyses . . . 245

Setting Up Analyses . . . 246
Disabling an Analysis . . . 247
Deleting an Analysis . . . 247
Specifying Order for Analyses . . . 248
Setting Up a Spectre Analysis . . . 249
Transient Analysis . . . 250
Time Step . . . 255
Algorithm . . . 255
State File . . . 259


© 2023 Cadence Design Systems, Inc.

Output . . . 260
EM/IR Output . . . 263
Fault . . . 264
Electrothermal . . . 264
Misc . . . 267
Transient Noise Analysis . . . 269
DC Analysis . . . 278
AC Analysis . . . 285
Noise Analysis . . . 288
S-parameter Analysis . . . 292
Transfer Function Analysis . . . 303
Sensitivity Analysis . . . 306
DC Match Analysis . . . 309
AC Match Analysis . . . 314
Stability Analysis . . . 323
Pole-Zero Analysis . . . 329
Loop Finder Analysis . . . 336
Other Spectre Analyses . . . 340
Setting Up an Analysis for UltraSim . . . 341
Setting Up an Analysis for AMS . . . 345
Setting Up an Analysis for HSPICE . . . 346
DC Analysis . . . 347
Transient Analysis . . . 349
AC Analysis . . . 350
Noise Analysis . . . 351
OP Analysis . . . 352
AC Match Analysis . . . 353
DC Match Analysis . . . 354
LSTB Analysis . . . 355
Setting Up EM/IR Analysis . . . 356
Specifying Basic Settings . . . 357
Specifying Technology Files . . . 360
Setting Up Voltus-Fi Options . . . 362
Setting Up the Solver Method and Time Window . . . 364
Reviewing Summary Information and Adding Customized Options . . . 366
Viewing Results . . . 368


© 2023 Cadence Design Systems, Inc.

Viewing the Layout Analysis . . . 371
EMIR Analysis Setup Form Description . . . 374
EMIR Analysis Setup – Basic Tab . . . 374
EMIR Analysis Setup – Techfile Tab . . . 379
EMIR Analysis Setup – Voltus-Fi Tab . . . 380
EMIR Analysis Setup – Solver Tab . . . 381
EMIR Analysis Setup – Summary Tab . . . 382
Setting Up EM/IR Analysis for AMS Simulator . . . 382
Running Advanced Analysis Simulations . . . 382

4
Specifying Output Data to be Saved or Plotted . . . 405
Using the Outputs Setup Pane . . . 406
Specifying Whether a Result is to be Saved or Plotted . . . 406
Type of Outputs to be Saved or Plotted . . . 407
Saving Output Expression . . . 408
Saving Output Signals . . . 414
Saving a Digital Bus . . . 416
Saving OCEAN or MATLAB Measurement . . . 418
Editing an OCEAN or a MATLAB Script File . . . 420
Writing a MATLAB Measure . . . 421
Creating Expression Outputs Using the Expression Builder . . . 423
Defining Specifications for an Output . . . 450
Saving Violation Filter . . . 453
Saving Operating Point Parameters . . . 453
Using MATLAB® Expressions in Outputs . . . 459
Deleting Outputs . . . 460
Adding a User-Defined Column . . . 460
Renaming a User-Defined Column . . . 461
Deleting a User-Defined Column . . . 461
Filtering and Searching in Outputs Setup Pane . . . 462
Importing and Exporting Outputs to a CSV File . . . 467
Structure of a CSV File . . . 467
Exporting Outputs . . . 468


© 2023 Cadence Design Systems, Inc.

Importing Outputs . . . 469
Specifying which Outputs to Save . . . 472
Saving Selected Voltages or Currents . . . 472
Saving All Voltages or Currents . . . 473
Expressions to be Measured Across Dimensions . . . 489
Environment Variables for PSFXL Output Format . . . 490
PSFXL Environment Variables for Site Administrators . . . 493

5
Working with Device Instance Parameters . . . 495

Creating Device Instance Parameters . . . 496
Changing the Value of Device Instance Parameters . . . 504
Sweeping Pcell Parameters . . . 505
Specifying Values of Device Instance Parameters . . . 506
Creating Matched Device Parameters . . . 506
Creating Ratio-Matched Device Parameters . . . 508
Deleting a Device Parameter . . . 510
Enabling and Disabling Parameters . . . 512
Adding Notes to Parameters . . . 512
Viewing Parameters by Property and Object Names . . . 514
Filtering Device Instance Parameters . . . 515
Creating Custom Device Filters . . . 517
Changing the Order of Device Parameters . . . 518
Disabling Callbacks on Swept Device Parameters . . . 519
Adding or Modifying Specifications for Parameters and Design Variables . . . 520
Specifying an Inclusion List of Values . . . 521
Specifying an Exclusion List of Values . . . 522
Specifying a Range of Values . . . 523
Specifying Center and Span . . . 525
Specifying Center and Span as a Percentage . . . 527
Deleting a Parameter Specification . . . 530
Creating Parameter Ranges . . . 531
Working with Parametric Sets . . . 535
Adding a Variable to a Parametric Set . . . 539
Removing a Variable from a Parametric Set . . . 540


© 2023 Cadence Design Systems, Inc.

Ungrouping Parametric Sets . . . 540
Enabling or Disabling Parametric Sets . . . 540

6
Working with Corners . . . 541
Opening the Corners Setup Form . . . 542
Adding Corners . . . 546
Specifying Temperature . . . 548
Specifying Values for Design Variables and Parameters . . . 548
Adding Model Files to a Corner . . . 550
Specifying Sections for Model Files . . . 553
Adding a Model Group to a Corner . . . 554
Creating a Model Group . . . 555
Saving and Loading Model Groups . . . 557
Adding Corners Tags . . . 557
Modifying Corner Values . . . 558
Working with Corners . . . 559
Renaming a Corner . . . 559
Adding Notes to a Corner . . . 560
Disabling and Enabling Corners . . . 560
Changing the Order of Corners . . . 564
Removing Corners Using the Setup Assistant . . . 565
Removing Corners Using the Corners Setup Form . . . 565
Managing Corners Using Editing Options . . . 567
Viewing Corner Settings . . . 569
Copying Corners . . . 569
Exporting Corners . . . 570
Importing Corners . . . 573
Setting Up a Default Set of Corners . . . 580
Running Resource Estimation for a Specific Corner . . . 581
Working with Corner Groups . . . 582
Creating a Corner Group . . . 582
Expanding a Corner Group . . . 585
Filtering Corners . . . 587
Saving Filters . . . 593


© 2023 Cadence Design Systems, Inc.

Deleting Saved Filters . . . 594
Enabling or Disabling Filtered Corners . . . 594
Resizing Corners . . . 594
Simulating Corner . . . 595

7
Running a Simulation . . . 597
Prerequisites to Run Simulation . . . 598
ADE Explorer Simulation Flow . . . 599
Setting up Job Policies . . . 600
Setting Up the Default Job Policy . . . 603
Specifying a Job Policy Name . . . 604
Optimizing a Single Point Run . . . 604
Specifying a Distribution Method . . . 605
Specifying Max Jobs . . . 606
Specifying an LBS Job Policy . . . 607
Specifying Job Timeouts . . . 610
Specifying Error Reporting Options . . . 611
Saving or Deleting a Job Policy . . . 612
Running Simulations With ICRP Job Control Mode . . . 614
Running Simulations With ICRP Job Control Mode . . . 615
Specifying Multiple Run Options . . . 616
Setting Up Resource Options . . . 616
Suspending and Resuming Simulation Runs . . . 621
Automatically Suspend Simulations When the Disk Space is Low . . . 621
Manually Suspend Simulations . . . 624
Starting a Simulation . . . 627
Simulation RunTime Improvements . . . 629
Performance Enhancements by Simulating Group of Points . . . 633
Interrupting or Stopping a Simulation . . . 635
Canceling Simulations for Selected Corners . . . 635
Viewing Run Preview . . . 639
Refreshing the Setup Summary and Points Table . . . 640
Enabling or Disabling Run Points in Points Table . . . 643
Using the Information Displayed on the Run Preview Tab . . . 644


© 2023 Cadence Design Systems, Inc.

Setting the Debugging Options . . . . . 647
Debugging Points . . . . . 650

8
Performing Monte Carlo Analysis . . . . . 657
Before Running a Monte Carlo Analysis . . . . . 658
Specifying Monte Carlo Run Options . . . . . 660
Performing a Standard Monte Carlo Run . . . . . 663
Performing Sensitivity Accuracy . . . . . 670
Performing Yield Verification . . . . . 673
High Yield Methods . . . . . 678
Creating Statistical Corners . . . . . 684
Running Confidence Interval - Autostop Method . . . . . 691
Specifying Instances for Monte Carlo Mismatch and Process Variation . . . . . 692
Selecting Schematic Instances for Applying Mismatch Variations . . . . . 695
Selecting Instances of Cellviews for Applying Mismatch Variations . . . . . 698
Selecting Subcircuit Instances for Applying Mismatch Variations . . . . . 701
Manually Specifying the Instances for Applying Mismatch Variations . . . . . 704
Modifying an Existing List of Selected Instances . . . . . 705
Deleting Instances Selected for Applying Mismatch Variations . . . . . 706
Running Advanced Methods . . . . . 708
Viewing Monte Carlo Results . . . . . 714
Managing Monte Carlo Results in the Yield View . . . . . 715
Expanding and Collapsing Output Data . . . . . 715
Viewing Data for a Specific Confidence Interval . . . . . 716
Filtering Out Data from the Yield View . . . . . 718
Generating Plots, Tables, Files, and Reports . . . . . 720
Plot/Print versus Iteration . . . . . 721
Printing Correlation Tables . . . . . 724
Plotting Histograms . . . . . 726
Plotting Scatter Plots . . . . . 738
Exporting Results to a CSV File . . . . . 741
Viewing Statistical Parameters for Monte Carlo Samples . . . . . 741
Displaying Statistical Parameters of Statistical Corners . . . . . 742


© 2023 Cadence Design Systems, Inc.

# 9
# Plotting and Printing . . . . . 745


Setting Plotting Mode . . . . . 747
Distributed Plot Commands . . . . . 755
Refreshing Graphs . . . . . 757
Performing Quick Plotting . . . . . 764
Using the Plot Output Commands . . . . . 767
Plotting the Current or Restored Results . . . . . 767
Plotting Parasitic Simulation Results . . . . . 769
Using the Direct Plot Commands . . . . . 770
Using the Direct Plot Main Form . . . . . 775
For DC . . . . . 775
For AC . . . . . 779
For Transient Results . . . . . 781
For Noise Results . . . . . 789
For Noise Separation . . . . . 792
For Stability Results . . . . . 795
For Pole Zero Results . . . . . 799
For Transfer Function . . . . . 802
For S-Parameter . . . . . 804
S-Parameter Probing . . . . . 807
For Loop Finder . . . . . 808
Printing Results . . . . . 811
Printing Components . . . . . 817
Printing DC Operating Points . . . . . 818
Printing Transient Operating Points . . . . . 818
Printing Model Parameters of Components . . . . . 819
Printing Noise Parameters of Nodes or Components . . . . . 819
Noise Summary . . . . . 820
Printing DC Mismatch Summary . . . . . 831
Printing AC Match Summary . . . . . 832
Printing Stability Summary . . . . . 834
Printing Pole Zero Summary . . . . . 836
Printing DC Node Voltages . . . . . 838
Printing Transient Voltages . . . . . 839



© 2023 Cadence Design Systems, Inc.

Printing Sensitivities . . . 839
Printing Capacitance Data . . . 839
Precision Control for Printing . . . 841
Using SKILL to Display Tabular Data . . . 842
Annotating Simulation Results . . . 843
Annotating DC Node Voltages . . . 844
Annotating DC Operating Points . . . 844
Annotating DC Currents . . . 844
Annotating Transient Voltages . . . 844
Annotating Transient Currents . . . 845
Annotating Transient Operating Points . . . 845
Annotating AC Operating Points . . . 846
Specifying the Data Directory for Labels . . . 846
Saving and Removing Annotated Labels . . . 848
Conditional Search for Results . . . 848
Form Field Descriptions . . . 851
Setting Plotting Options . . . 851
XF Results . . . 852
S-Parameter Results . . . 853
Circuit Conditions . . . 856

10
Real-Time Tuning in Simulations . . . 861
Opening the Real-Time Tuning Assistant . . . 862
RTT Graphical User-Interface . . . 864
Working with the RTT Assistant . . . 865
Backannotating Variables and Parameters from RTT . . . 872
Some Useful Points . . . 874
Viewing Results in Virtuoso Visualization and Analysis XL . . . 876
Viewing Waveform Info-Balloons on Schematic . . . 879

11
Viewing Results . . . 891
Viewing the Simulation Results . . . 892
Hiding and Showing Data on the Results Tab . . . 894


© 2023 Cadence Design Systems, Inc.

Hiding Test Details . . . 896
Hiding Specification Details . . . 896
Hiding Corner Results . . . 896
Hiding Signals . . . 897
Hiding Measured Result Values . . . 897
Hiding Minimum and Maximum Values . . . 897
Showing Only Errors . . . 897
Filtering Results Using Column Filters . . . 897
Showing the Starting Point . . . 898
Showing Fixed Parameters for a Design Point . . . 898
Viewing Netlist for Results . . . 899
About Results Views . . . 900
Detail . . . 900
Detail - Transpose . . . 901
Optimization . . . 902
Status . . . 903
Summary . . . 904
Yield . . . 905
Capacitance . . . 907
Using the Detail - Transpose View . . . 907
Using the Status View . . . 908
Freezing Columns . . . 908
Sorting Results in the Detail and Detail-Transpose Views . . . 910
Working with Plotting Templates . . . 911
Backannotating from ADE Explorer Results . . . 918
Re-evaluating Expressions and Specifications . . . 920
Viewing Specification Results in the Results Tab . . . 921
Comparing S-Parameter Data . . . 923
Viewing Results in Error Mode . . . 925
Viewing Results for Passivity Mode . . . 927
Interpolating S-Parameter Data . . . 929
Working with Datasheets . . . 931
Creating a Datasheet for a Checkpoint . . . 931
Displaying Customized Waveform Images in the Datasheet . . . 940
Opening a Datasheet . . . 940
Customizing the Datasheet Format and Structure . . . 941


© 2023 Cadence Design Systems, Inc.

12
Performing Reliability Analysis . . . . . 943
Introduction . . . . . 944
Specifying Reliability Options . . . . . 944
Specifying Aging Model Files . . . . . 947
Specifying Reliability Options . . . . . 949
Migrating Reliability Setup from ADE L . . . . . 954
Running the Reliability Simulation . . . . . 956
Viewing the Reliability Simulation Results . . . . . 957
Device Lifetime and Degradation Results . . . . . 957
Model Parameter Changes Results . . . . . 958
Viewing the Reliability Aged Netlist . . . . . 959
Annotating Simulation Results to the Schematic . . . . . 959
Saving Reliability Options . . . . . 961
Loading Reliability Options . . . . . 962

13
Performing Circuit and Device Checks . . . . . 965
An Introduction to Checks and Asserts . . . . . 966
Checks . . . . . 966
Asserts . . . . . 966
Dependency on Simulators . . . . . 967
Viewing Checks and Asserts in the Setup Assistant . . . . . 967
Adding Checks and Asserts . . . . . 968
Adding Checks using the Checks/Asserts Assistant . . . . . 968
Copying Checks or Asserts . . . . . 975
Importing Asserts from Text Files Included in the Setup . . . . . 977
Filtering Violations . . . . . 980
Running Simulations and Viewing Violations . . . . . 983
Specifying the Global Device Check Options . . . . . 984

14
Helping Simulations to Converge . . . . . 989
Commands for Forcing Convergence . . . . . 989


© 2023 Cadence Design Systems, Inc.

Node Set . . . 990
Initial Conditions . . . 990
Force Node . . . 991
HspiceD Convergence Aids . . . 991
Selecting Nodes and Setting their Values . . . 992
Releasing Voltages . . . 993
Changing Voltages . . . 994
Loading an Existing Convergence Setup . . . 995
Highlighting Set Nodes . . . 996
Storing a Solution . . . 996
Restoring a Solution for Spectre . . . 997

15
Working with AMS Simulator . . . 999

Creating a Configuration Cellview . . . 1001
Setting Up the Default Simulator for a Configuration . . . 1002
Setting Up a Config Cellview Using Virtuoso Schematic Editor or Virtuoso Text Editor 1003
ADE Explorer and AMS Integration . . . 1010
Setting up AMS Options . . . 1012
Working with Global Signals in AMS . . . 1026
Additional Arguments Form . . . 1043
Setting up Environment Options . . . 1044
About the AMS Unified Netlister . . . 1071
Benefits of AMS UNL . . . 1074
Things to Know When Using the AMS UNL Flow . . . 1074
Migrating to AMS UNL . . . 1081
Features of AMS UNL . . . 1083
Exporting IP Blocks in AMS UNL . . . 1091
Limitations of AMS UNL . . . 1106
Setting Up the AMS Netlister Options . . . 1112
Process-Based Save and Restart (PBSR) flow . . . 1114
Saving Simulation Snapshots at Different Time Points . . . 1115
Saving Simulation Snapshots Periodically . . . 1116
Pre-requisites for Restarting a simulation from a Saved Snapshot . . . 1117


© 2023 Cadence Design Systems, Inc.

Restarting a Simulation from a Saved Snapshot . . . 1117
Netlist and Run Options Form . . . 1119
Simulating the Design Using the xrun Command . . . 1126
Debugging Verilog-A Instances in AVUM flow . . . 1130
Saving or Plotting Selected Voltages or Currents for AMS Simulation . . . 1131
Adding a Node or Terminal to a Set . . . 1135
DSPF-in-the-Middle Flow for AMS Simulations . . . 1136
Using the runams Command . . . 1137
runams Command Options . . . 1138
How to Use runams Options . . . 1155
runams Command Examples . . . 1157
Viewing the Output Log for AMS . . . 1159
Viewing the Error Explanation for AMS . . . 1160
Using the SimVision Debugger . . . 1161
Setting Up a Cross Selection Menu to Launch the Post Simulation Analysis Form 1163
Display Partition . . . 1165
Default Digital Discipline Selection . . . 1169

16
Working with HSPICE Simulator . . . 1175
Introduction to HSPICE Direct Simulator . . . 1176
Libraries . . . 1177
HspiceD Features . . . 1179
Model Libraries . . . 1179
Distributed Processing Support . . . 1179
Running Analyses . . . 1179
Passing Command-Line Options . . . 1180
Specifying HSPICE Options . . . 1181
Output Log . . . 1184
Convergence Aids . . . 1185
Results . . . 1185
Control Mapping of Nets . . . 1187
Setting up Reliability Option . . . 1187


© 2023 Cadence Design Systems, Inc.

17
Simulating Multi-Technology Designs and Defining Locally Scoped Models and Options . . . 1191
Enabling Locally Scoped Models and Options . . . 1192
Enabling Local Scoping for a Cell . . . 1194
Enabling Local Scoping for an Instance . . . 1198
Disabling Local Scoping . . . 1202
Limiting the View by Library or Instance . . . 1203

18
auLvs Netlisting . . . 1205
Using auLvs . . . 1205
Customization Using the .simrc File . . . 1205
Related Documentation on auLvs . . . 1206

19
Working with UltraSim Simulator . . . 1207
Specifying UltraSim Simulator Interface . . . 1208
Specifying UltraSim Options . . . 1210
UltraSim Power Network Solver . . . 1224
UltraSim Interactive Simulation Debugging . . . 1228
Saving Outputs for UltraSim Simulations . . . 1230
Saving Selected Voltages or Currents . . . 1233

A
auCdl Netlisting . . . 1235
What Is auCdl and Why Do You Need It? . . . 1235
Licensing Requirements . . . 1236
Running auCdl . . . 1236
Running auCdl from within DFII . . . 1236
Running auCdl from the Command-Line . . . 1237
Creating a config view for auCdl . . . 1240
How to include partial netlist file in SUBCKT calls . . . 1241


© 2023 Cadence Design Systems, Inc.

Generating individual SUBCKT netlists . . . 1242
Customization Using the .simrc File . . . 1244
auCdl-Specific Parameters . . . 1244
View List, Stop List, Netlist Type, and Comments . . . 1250
Preserving Devices in the Netlist . . . 1250
Removing Devices in the Netlist . . . 1251
Printing CDL Commands . . . 1251
Defining Power Node and Ground Node . . . 1251
Support for Global Power and Ground Signals from CDL UI . . . 1252
Evaluating Expressions . . . 1252
NLP Expressions . . . 1252
Mapping Global Pins . . . 1253
Renaming Cell Names . . . 1254
Renaming Pcell Subcircuits . . . 1254
Customizing Bulk Node Search . . . 1254
Support for HED Features . . . 1256
Custom Netlisting Procedures . . . 1257
ansCdlSubcktCall . . . 1257
ansCdlCompPrim . . . 1258
ansCdlCompParamPrim . . . 1258
ansCdlSpecParamPrim . . . 1259
ansCdlSubcktCallExtended . . . 1259
ansCdlHnlPrintInst . . . 1260
Netlist Customization Using netlistProcOpts . . . 1266
Blackbox Netlisting . . . 1270
Additional Customizations . . . 1274
Automatically Including a Partial Netlist File within the .SUBCKT Definition for the Top or Mid-Level Cells in your Design . . . 1274
Including a ROM-Insert Netlist Automatically Into the auCdl Netlist . . . 1277
PININFO for Power and Ground Pins . . . 1278
.PARAM Statement . . . 1278
Specifying Terminal Order for auCDL Netlisting in the Edit CDF Form . . . 1279
Specifying the Terminal Order for Terminals in .simrc . . . 1280
Rules followed to Print Terminal Order during Netlisting . . . 1282
Changing the Pin Order . . . 1286
Arranging the Terminal Ordering . . . 1287


© 2023 Cadence Design Systems, Inc.

Notification about Net Collision . . . 1287
Making a Stop Cell at Subcircuit Level . . . 1290
Printing Empty Subcircuits . . . 1290
Passing Parameter . . . 1293
Netlisting the Area of an npn . . . 1293
CDF Simulation Information for auCdl . . . 1294
Device CDF Values . . . 1294
Netlist Examples . . . 1298
What is Different in the 4.3 Release . . . 1299
Complete Example . . . 1301

B
Environment Variables . . . 1305

Simulation Environment . . . 1307
LBS Distributed Processing Environment . . . 1309
Simulator-Specific Environment Variables . . . 1309
Environment Variables for Distributed Processing . . . 1310
Environment Variables for Distributed Processing with LBS . . . 1318
adexl.monte . . . 1324
additionalNetlistOptions . . . 1325
applySaveOptionsToNetlist . . . 1326
createStatisticalCornerType . . . 1327
enableMonteCarloOverStatisticalCorners . . . 1328
incrementalUpdate . . . 1329
iterationUpdates . . . 1330
minGroupSizeSplitAcrossIdleJobs . . . 1331
numberOfPointsToview . . . 1332
samplingMethod . . . 1333
saveProcessOptionDefaultValue . . . 1334
saveMismatchOptionDefaultValue . . . 1335
saveSimulationData . . . 1336
AMS . . . 1337
ams.emirOpts . . . 1338
ams.envOpts . . . 1339
amsIEsList . . . 1339


© 2023 Cadence Design Systems, Inc.

autoConfigNameForConfigCreation . . . 1340
case_sensitive . . . 1341
connectRulesList . . . 1341
constraintListForConfigCreation . . . 1342
. . . 1343
defaultVsupForVAR . . . 1343
disableCompileVAAsVAMS . . . 1343
disableRunModeInDP . . . 1344
enableChangeSpectrePath . . . 1344
filesOnIrunCmdLineHDL . . . 1345
ipAddIpLabelAsPrefix . . . 1345
ipExportHDLFiles . . . 1346
irunIncDirHDL . . . 1346
libDirsHDL . . . 1347
libFilesHDL . . . 1347
libraryListForConfigCreation . . . 1348
maxNumSnapShots . . . 1348
optsFileHDL . . . 1349
saveAllSnapShots . . . 1350
snapShotBaseName . . . 1350
snapShotNameTimeUnit . . . 1351
snapShotSaveMode . . . 1351
spectrePath . . . 1352
stopListForConfigCreation . . . 1352
strobeTime . . . 1353
useIeSetup . . . 1353
viewListForConfigCreation . . . 1354
switchViewList . . . 1355
enableAMSdebug . . . 1355
ipPreserveEnvVars . . . 1356
ipEnableDutCell . . . 1356
ipDutCell . . . 1357
ipDutLib . . . 1357
ipDutView . . . 1358
ieUseUcmAsDefault . . . 1359
ams.netlisterOpts . . . 1360


© 2023 Cadence Design Systems, Inc.

unlAnalogOnly . . . 1360
asimenv . . . 1361
ignoreUnsetDesVars . . . 1361
asimenv.monte . . . 1362
highPrecision . . . 1362
precisionDigits . . . 1362
asimenv.netlist . . . 1363
issueInfoMsgOnCellRename . . . 1363
auCore.misc . . . 1364
doExactTermMatch . . . 1364
auCore.selection . . . 1365
alwaysFlattenNames . . . 1365
HspiceD . . . 1366
license . . . 1367
maestro.cpupdtr . . . 1368
maestro.distribute . . . 1369
maestro.emir . . . 1370
showEnableRFReferenceFlowUI . . . 1370
maestro.explorer . . . 1371
maestro.gui . . . 1375
confirmBeforeTestOrCornerDelete . . . 1375
nameDisplayWidthInDataView . . . 1375
resultsFilteringColumnsList . . . 1376
showFilterInDataView . . . 1376
useExpressionBuilder . . . 1377
usePlottingTemplate . . . 1377
yieldViewMaxOutputsShowExpanded . . . 1378
saveImplicitSignalsDuringRun . . . 1378
hideShowMultiProcess . . . 1379
showUnitsInResults . . . 1380
textColorForSpecFailInResults . . . 1381
textColorForSpecNearInResults . . . 1382
textColorForSpecPassInResults . . . 1382
showNativeRelxpertSwitch . . . 1383
showRedundantPointsinDT . . . 1385
defaultParametersAllowList . . . 1386


© 2023 Cadence Design Systems, Inc.

defaultParametersAssistantFilter . . . 1387
enableGroupByTagsForVars . . . 1388
enableGroupByTagsForCorners . . . 1389
disableInfoAnalysisOutputWhenUIDisabled . . . 1390
disableHighPerformanceSimulationEnableMS . . . 1391
hideStringResultsFailures . . . 1392
defaultParametersAllowList . . . 1393
showRemoveTmpADEDirButton . . . 1394
showMSOptionsInSpectreXTab . . . 1395
deleteUnusedProbes . . . 1396
disableHighPerformanceSimulationSpectreXOption . . . 1397
donotExpandNameForparam_file . . . 1398
showNoiseSummaryinTableWidget . . . 1399
maestro.license . . . 1400
maestro.mpt . . . 1401
maestro.plotting . . . 1403
distributedPlotEnabled . . . 1404
distributedPlotEnableCIW . . . 1405
distributedPlotJobPolicy . . . 1406
distributedPlotLoadCdsInit . . . 1407
distributedPlotLingerTime . . . 1408
distributedPlotPolicyNameFilter . . . 1409
distributedPlotEnableNonSupportedPlotWarning . . . 1410
quickPlotGlobalWriteDisable . . . 1411
maestro.polo . . . 1412
dspfBindingSummary . . . 1412
dspfBindingToCell . . . 1413
mspsSkipMappingForExtractedViews . . . 1414
netlistSmartViewDSPFOnRecreateNetlist . . . 1415
supportDspfMasterOption . . . 1416
plotSensitivityDataInSeparateSubwindows . . . 1417
maestro.simulation . . . 1418
maestroViewEnvName . . . 1423
cancelRunAfterRetries . . . 1424
maestro.stimuli . . . 1426
maestro.stimuliTool . . . 1427


© 2023 Cadence Design Systems, Inc.

maestro.variables . . . 1428
maestro.window . . . 1431
Spectre . . . 1432
AC Match Analysis Environment Variables . . . 1433
Spectre.emirOpts . . . 1435
advancedTable . . . 1436
anaType . . . 1437
autoLoadVfiFile . . . 1438
dspfListBox . . . 1439
dynamicAnaSummaryBox . . . 1440
editSummaryInfoTable . . . 1441
emDataFile . . . 1442
emOnlyIctFile . . . 1443
enableEMIRAna . . . 1444
hideAnalysesItems . . . 1445
ictFile . . . 1446
includeFile . . . 1447
layerName . . . 1448
netName . . . 1449
qrcTechFile . . . 1450
short . . . 1451
shortResLayer . . . 1452
solverMethod . . . 1453
solverMethodTable . . . 1454
spgsAdvIRDrop . . . 1455
spgsFilterMoscap . . . 1456
staticAnaSummaryBox . . . 1457
summaryListBox . . . 1458
techFileMode . . . 1459
timeWindowListBox . . . 1460
Spectre.envlpOpts . . . 1461
sweepmethod . . . 1461
Spectre.envOpts . . . 1462
case_sensitive . . . 1462
lsusp . . . 1462
lsuspEnvOverridesState . . . 1463


© 2023 Cadence Design Systems, Inc.

setTopLevelAsSubckt . . . 1464
Spectre.hb . . . 1465
dynchoice . . . 1465
Spectre.opts . . . 1466
enable_dcsweep_op_info . . . 1466
tranFilterExtreme . . . 1467
Spectre.pss . . . 1468
Spectre.tran . . . 1469
dynamicChoice . . . 1469
dynamicParameter . . . 1469
dynParamChoice . . . 1470
dynParamInput . . . 1470
dynParamSetChoice . . . 1471
dynParamSetInput . . . 1472
Spectre X . . . 1473
spectreXAdvanced . . . 1473
spectreXPreset . . . 1474
spectreXMpOption . . . 1474
spectreXEnablePostLayout . . . 1475
spectreXAccuracyScale . . . 1475
spectreXHosts . . . 1476
spectreXMpThreads . . . 1476
spectreXMtOption . . . 1477
spectreXNumThreads . . . 1477
spectreXMTPOption . . . 1478
uniMode . . . 1478
spectreVerilog . . . 1479
Ultrasim . . . 1480
voltus_fi.results . . . 1481
vfiEnableGUI . . . 1481
Real Time Tuning Environment Variables . . . 1482
Environment Variables . . . 1487


© 2023 Cadence Design Systems, Inc.

# C
# Environment Variables for Spectre Simulator Options Form . . 1501

spectre.opts . . . . . 1502
Simulator Options - Main . . . . . 1505
checklimitskipinsts . . . . . 1505
generalnoiseinst . . . . . 1505
generalnoiseinstonoff . . . . . 1505
highvoltage . . . . . 1506
iabstol . . . . . 1506
multithread . . . . . 1507
noiseOffType . . . . . 1507
noiseOnType . . . . . 1508
nthreads . . . . . 1508
reltol . . . . . 1509
residualtol . . . . . 1509
temp . . . . . 1509
tempeffects . . . . . 1510
tnom . . . . . 1510
vabstol . . . . . 1511
Simulator Options - Algorithm . . . . . 1512
convdbg . . . . . 1512
dc_pivot_check . . . . . 1512
dptran_gmethod . . . . . 1512
gmethod . . . . . 1513
gmin . . . . . 1513
gmin_check . . . . . 1514
gmindc . . . . . 1514
homotopy . . . . . 1515
icpriority . . . . . 1515
ilimit . . . . . 1516
nonconv_topnum . . . . . 1516
pivabs . . . . . 1516
pivotdc . . . . . 1517
pivrel . . . . . 1517


© 2023 Cadence Design Systems, Inc.

preorder . . . 1518
rabsclamp . . . 1518
rabsshort . . . 1519
rebuild_matrix . . . 1519
rforce . . . 1520
rthresh . . . 1520
try_fast_op . . . 1521
Simulator Options - Component . . . 1522
approx . . . 1522
auto_minductor . . . 1522
vth_vdsmin . . . 1522
ivthl . . . 1523
ivthn . . . 1523
ivthp . . . 1524
ivthw . . . 1524
imacromodels . . . 1524
maxrsd . . . 1525
nport_default_passivity . . . 1525
nportcompress . . . 1526
nportcompressfiledir . . . 1526
nportirfiledir . . . 1527
nportirreuse . . . 1527
nportunusedportgmin . . . 1528
nportunusedportrmin . . . 1528
nportbbsversion . . . 1529
nportbbsfittedfiledir . . . 1529
minr . . . 1530
scale . . . 1530
scalefactor . . . 1531
scalem . . . 1531
tmevthmod . . . 1532
vthmod . . . 1532
Simulator Options - Check . . . 1533
checklimitdest . . . 1533
checklimitfile . . . 1533
checklimitskipfile . . . 1533


© 2023 Cadence Design Systems, Inc.

checklimitskipsubs . . . 1534
diagnose . . . 1534
iccapcheck . . . 1535
gnshorts . . . 1535
redefinedparams . . . 1536
opptcheck . . . 1536
topcheck . . . 1537
Simulator Options - Annotation . . . 1538
audit . . . 1538
cols . . . 1538
colslog . . . 1538
debug . . . 1539
digits . . . 1539
error . . . 1540
info . . . 1540
inventory . . . 1540
maxnotes . . . 1541
maxnotestologfile . . . 1541
maxwarns . . . 1542
maxwarnstologfile . . . 1542
notation . . . 1543
note . . . 1543
narrate . . . 1544
printstep . . . 1544
simstat . . . 1544
title . . . 1545
warn . . . 1545
Simulator Options - Miscellaneous . . . 1547
additionalArgs . . . 1547
ahdllint . . . 1547
ahdllint_maxwarn . . . 1547
ahdllint_on . . . 1548
flow . . . 1548
quantities . . . 1549
sensbinparam . . . 1549
sensfile . . . 1550


© 2023 Cadence Design Systems, Inc.

sensfileonly . . . . . 1550
sensformat . . . . . 1550
senstype . . . . . 1551
value1 . . . . . 1551

D
Environment Variables for Spectre Analyses Options Form . . 1553

spectre.ac . . . . . 1554
center . . . . . 1555
contriblist . . . . . 1555
dec . . . . . 1556
designVar . . . . . 1556
dev . . . . . 1557
deviceParam . . . . . 1557
enable . . . . . 1557
fim_out . . . . . 1558
flin_out . . . . . 1558
freq . . . . . 1559
incrType . . . . . 1559
lin . . . . . 1560
log . . . . . 1560
maxharm_nonlin . . . . . 1560
mod . . . . . 1561
modelParam . . . . . 1561
out1 . . . . . 1562
out2 . . . . . 1562
outType . . . . . 1562
rangeType . . . . . 1563
rf1_freq . . . . . 1563
rf1_src . . . . . 1564
rf2_freq . . . . . 1564
rf2_src . . . . . 1564
rfdbm . . . . . 1565
rfdbm2 . . . . . 1565


© 2023 Cadence Design Systems, Inc.

rfdbm3 . . . 1566
rfmag . . . 1566
rfmag2 . . . 1566
rfmag3 . . . 1567
span . . . 1567
special . . . 1568
srcType . . . 1568
start . . . 1568
step . . . 1569
stepTypeLin . . . 1569
stepTypeLog . . . 1570
stop . . . 1570
sweep . . . 1571
useDiscrete . . . 1571
useFile . . . 1571
values . . . 1572
valuesFile . . . 1572
spectre.acOpts . . . 1574
additionalParams . . . 1575
annotate . . . 1575
force . . . 1575
nestlvl . . . 1576
oppoint . . . 1576
prevoppoint . . . 1577
readforce . . . 1577
readns . . . 1578
restart . . . 1578
save . . . 1578
skipdc . . . 1579
write . . . 1579
writefinal . . . 1580
spectre.dc . . . 1581
center . . . 1582
dec . . . 1582
designVar . . . 1582
dev . . . 1583


© 2023 Cadence Design Systems, Inc.

deviceParam . . . 1583
enable . . . 1584
hysteresis . . . 1584
incrType . . . 1585
lin . . . 1585
log . . . 1585
mod . . . 1586
modelParam . . . 1586
rangeType . . . 1587
saveOppoint . . . 1587
span . . . 1588
start . . . 1588
step . . . 1588
stepTypeLin . . . 1589
stepTypeLog . . . 1589
stop . . . 1590
sweep . . . 1590
useDiscrete . . . 1591
useFile . . . 1591
values . . . 1591
valuesfile . . . 1592
spectre.dcOpts . . . 1593
additionalParams . . . 1594
annotate . . . 1594
captab . . . 1595
check . . . 1595
detail . . . 1596
emirfile . . . 1596
emirformat . . . 1597
filename . . . 1597
force . . . 1598
homotopy . . . 1598
maxiters . . . 1599
maxsteps . . . 1599
nestlvl . . . 1600
oppoint . . . 1600


© 2023 Cadence Design Systems, Inc.

print . . . 1601
readforce . . . 1601
readns . . . 1602
restart . . . 1602
save . . . 1603
saveopinfinal . . . 1603
sort . . . 1604
threshold . . . 1604
write . . . 1605
writefinal . . . 1605
spectre.outputs . . . 1606
subcktoppoint . . . 1606
savePcell . . . 1606
saveBySubcktInstances . . . 1607
spectre.relxOpts . . . 1608
agingPointsType . . . 1608
checknegagingclamp . . . 1609
checknegagingtype . . . 1609
degCheckException . . . 1610
degSortType . . . 1610
degSortValue . . . 1611
enableDegSort . . . 1612
enableOutputDevDegInFile . . . 1612
enableRelXTran . . . 1613
enableSelfheating . . . 1613
gaAgePoints . . . 1614
gaAgePointsUnit . . . 1614
gaAgeStepStart . . . 1615
gaAgeStepStop . . . 1615
gaAgeStepTotalStep . . . 1615
gaAgeStepType . . . 1616
gaAgeStepUnit . . . 1617
gaMode . . . 1617
gaSaveResults . . . 1618
gaSaveSome . . . 1618
mindtemp . . . 1619


© 2023 Cadence Design Systems, Inc.

mosAgeDomain . . . 1619
mosAgeMethod . . . 1620
mosAgeModelFilePath . . . 1620
mosAgingTime . . . 1620
mosAgingTimeUnit . . . 1621
mosCalcLifetime . . . 1621
mosDeltad . . . 1622
mosHciAccuracy . . . 1622
mosHciIdMethod . . . 1623
mosHciOpMethod . . . 1623
mosMaskDev . . . 1624
mosMaskDevInclude . . . 1625
mosMaskDevInstance . . . 1625
mosMinAge . . . 1626
mosMode . . . 1626
mosOutputDevDegInFilePath . . . 1627
otherRXOptions . . . 1627
otherODDOptions . . . 1628
relxTranStart . . . 1629
relxTranStop . . . 1629
reportModelParamChanged . . . 1629
uriDebugMode . . . 1630
uriLibs . . . 1630
uriMode . . . 1631
Environment Variables for Reliability Analysis . . . 1632

E
Migrating From ADE L to ADE Explorer . . . 1635
Migrating an ADE L Setup to ADE Explorer . . . 1636
Migrating ADE L Setup Using Load State Form . . . 1636
Migrating an ADE L State by Opening the Cellview in ADE Explorer . . . 1640
Migrating ADE L Setup Using SKILL . . . 1642
Highlights of ADE Explorer . . . 1644
Customizing Menus in ADE Explorer . . . 1646
Reusing Changes from the .menu File . . . 1646


© 2023 Cadence Design Systems, Inc.

Customizing Menus Using a Trigger in .cdsinit . . . 1646
New Menus . . . 1647

F
ADE Explorer Frequently Asked Questions . . . 1649
Why does Zij show large values sometimes? . . . 1649
How to run more than 10 OCEAN sessions at once? . . . 1649
How to get the list of output signals and expressions set using SKILL? . . . 1650
How to read and load design variables from a text file to ADE L window? . . . 1651
Why does the RTT assistant not evaluate expressions if the output format is set to SST2? . . . 1652
Why do the schematic annotations not get updated automatically when a new simulation is run? . . . 1653

G
Cadence© Load Balancing Software (LBS) Distribution Method . . . 1655
Introduction to LBS Mode . . . 1655
System Administration of Distributed Jobs with LBS . . . 1656
Common LBS Setup Requirements . . . 1656
Additional Requirements . . . 1659
Setting up DRMS in LBS Distribution Method . . . 1660
System Requirements for cdsqmgr Setup . . . 1660
System Requirements for LSF Setup . . . 1670
System Requirements for SGE Setup . . . 1675
Validating the LBS Distributed Processing Setup . . . 1676
Submitting Simulation Jobs in LBS Distribution Method . . . 1677
Submitting Distributed Processing Jobs . . . 1677
Submitting Distributed Processing Jobs in OCEAN . . . 1678
Submitting Distributed Jobs in LBS using the Command Mode . . . 1680
Monitoring Distributed Jobs in LBS Method . . . 1681
Debugging an LBS Distributed Processing Setup . . . 1681
Debugging LBS with cdsqmgr . . . 1681
Debugging LBS with Third-Party DRMS . . . 1682
Troubleshooting FAQs . . . 1683


© 2023 Cadence Design Systems, Inc.

# 1

# Getting Started with ADE Explorer

## Introducing Virtuoso ADE Explorer

Virtuoso® ADE Explorer provides a single-testbench platform that you can use to define the simulation settings and run simulations based on the specified settings. You need to specify the following essential settings in the Virtuoso® ADE Explorer Setup before running a simulation—design variables, temperature, simulator, analysis, model files, and other setup files. You can also define the various outputs that are to be saved or plotted during a simulation run. By default, Spectre simulator is used to run simulations in ADE Explorer; however, you can choose other simulators as well, such as HSPICE, AMS, UltraSim.

When you run a simulation, the results are saved in a predefined formats and can be used by other tools such as the Virtuoso Visualization and Analysis XL for further analysis. If there are signals in the simulation results, they are by default plotted in the Virtuoso Visualization and Analysis XL Graph window.

Virtuoso® ADE Explorer is a part of the Virtuoso Design Framework II (DFII) environment, which is the foundation on which a wide range of Cadence tools are built. Using this architecture, you can go from one tool to another without tedious data conversion. The consistent user interface makes it convenient to apply your knowledge of one Cadence tool to many other Cadence tools. The design framework II environment is an open system. You can integrate third party tools and enter your own design data with industry-standard EDIF and Virtuoso® GDSII Stream formats. Circuit simulators can be integrated using OSS.

The Virtuoso® ADE Explorer provides the following key features:

- Consistent user-interface


© 2023 Cadence Design Systems, Inc.

- Display and analyze simulation outputs
- Real-time tuning of design
- View annotations
- Create, view, and run corners
- Add, view, and run specifications for outputs
- Set up and run sweeps
- Set up and run Monte Carlo simulation
- Perform device checks
- Define and run assertions

This chapter describes the following features that you can use to simulate the analog design using ADE Explorer:

- Opening Virtuoso ADE Explorer
- Virtuoso ADE Explorer Graphical User Interface
- Switching Between ADE Explorer and ADE Assembler

Video

For an overview of ADE Explorer, see the Getting Started with Virtuoso ADE Explorer video on Cadence Online Support.

# Licensing Requirements

The Virtuoso ADE Explorer requires one of the following licenses in the given order of precedence:

- `95250`, Virtuoso ADE Explorer
- `95260`, Virtuoso ADE Assembler

For information about licensing in the Virtuoso Studio design environment, see *Virtuoso Software Licensing and Configuration Guide*.


© 2023 Cadence Design Systems, Inc.

# Opening Virtuoso ADE Explorer

When you open the Virtuoso ADE Explorer for the first time, you first need to create a new `maestro` cellview to open your design in Explorer.

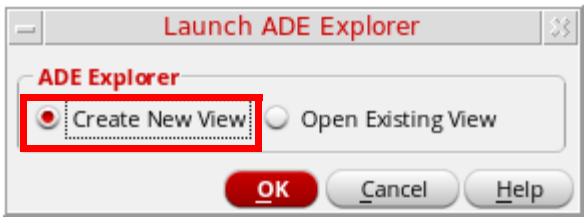
You can use one of the following methods to create maestro cellview and open ADE Explorer:

- Opening ADE Explorer From CIW
- Opening ADE Explorer From Virtuoso Schematic Editor
- Opening ADE Explorer Using Schematic Editor Workspace
- Virtuoso ADE Explorer Graphical User Interface

## Opening ADE Explorer From CIW

Follow the steps listed below to create a new or open an existing maestro view through Command-Interpreter Window (CIW):

1. In CIW, choose *Tools – ADE Explorer*. The *Launch ADE Explorer* form appears.

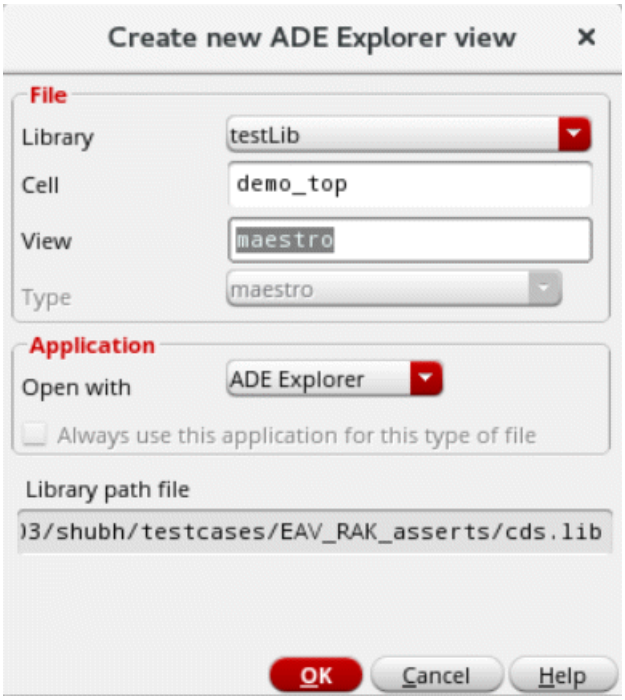



2. In this form, select the *Create New View* option if you are opening ADE Explorer for the first time. If you already have an maestro (ADE Explorer) view setup, select *Open Existing View*.

   **Note:** You can also open a design saved in Analog Design Environment L (ADE L) in Explorer. For more information, see Appendix E, “Migrating From ADE L to ADE Explorer”.


© 2023 Cadence Design Systems, Inc.

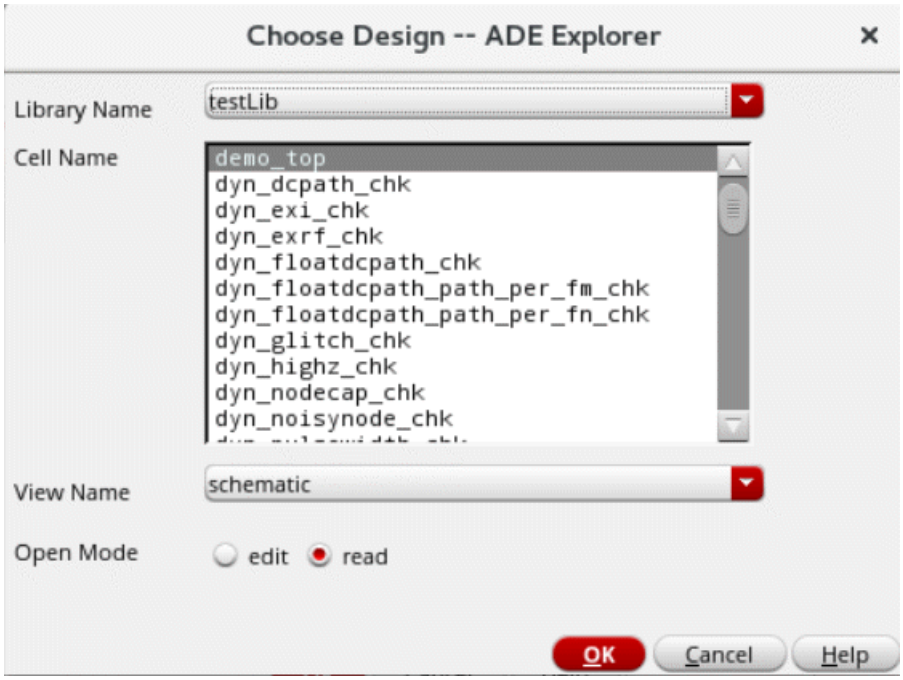
When you choose to create a new view, the *Create new ADE Explorer view* form appears as shown below.




**Note:** By default, the *Application* drop-down displays ADE Explorer as the application name; however, if you have the Assembler license, you can also select *Assembler* from this drop-down list to launch the design in Assembler.


© 2023 Cadence Design Systems, Inc.

3. Click *OK*. The *Choose Design – – ADE Explorer* form appears as shown below.

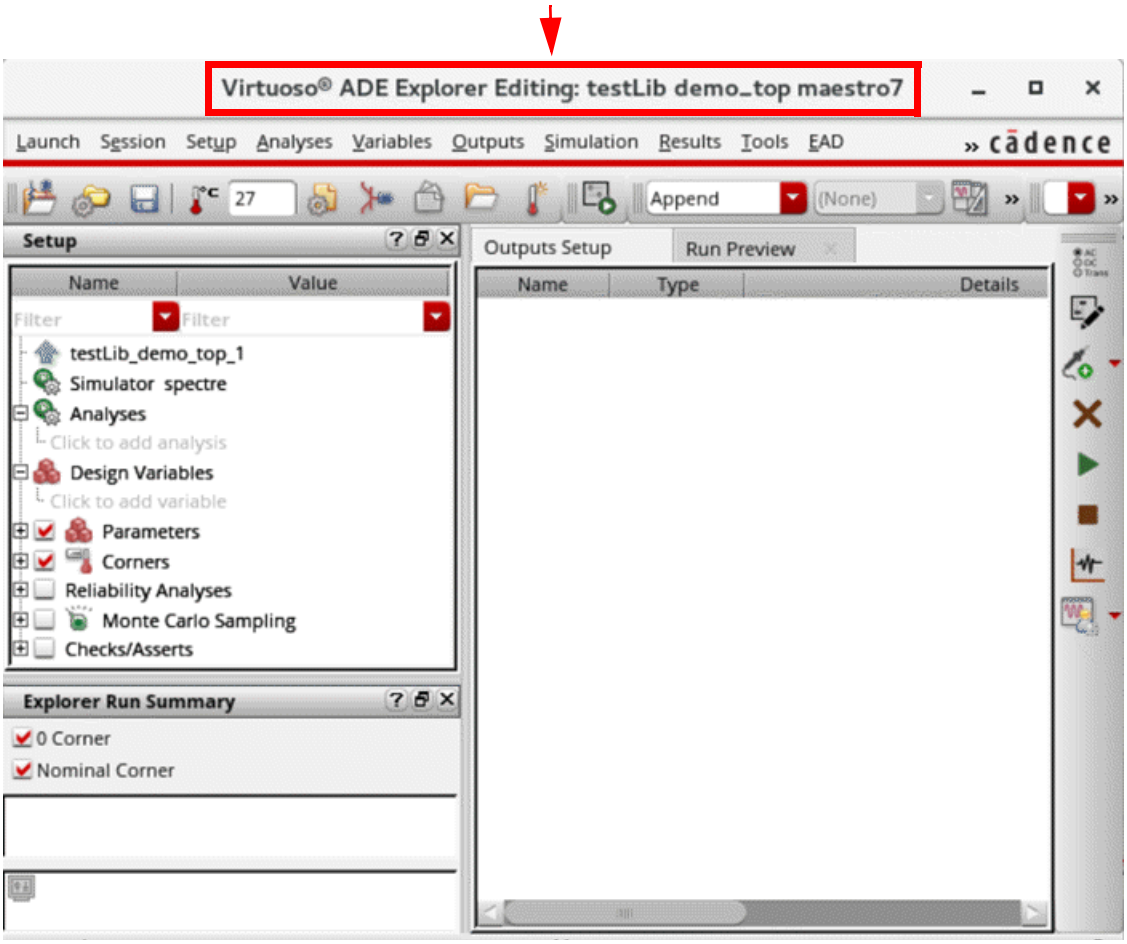



4. In this form, specify the library, cell name and view name. Also, specify whether you want to open the ADE Explorer in read or edit mode.


© 2023 Cadence Design Systems, Inc.

5. Click *OK*. The *Virtuoso ADE Explorer* window appears as shown below. You can now set up and run simulations for this design.

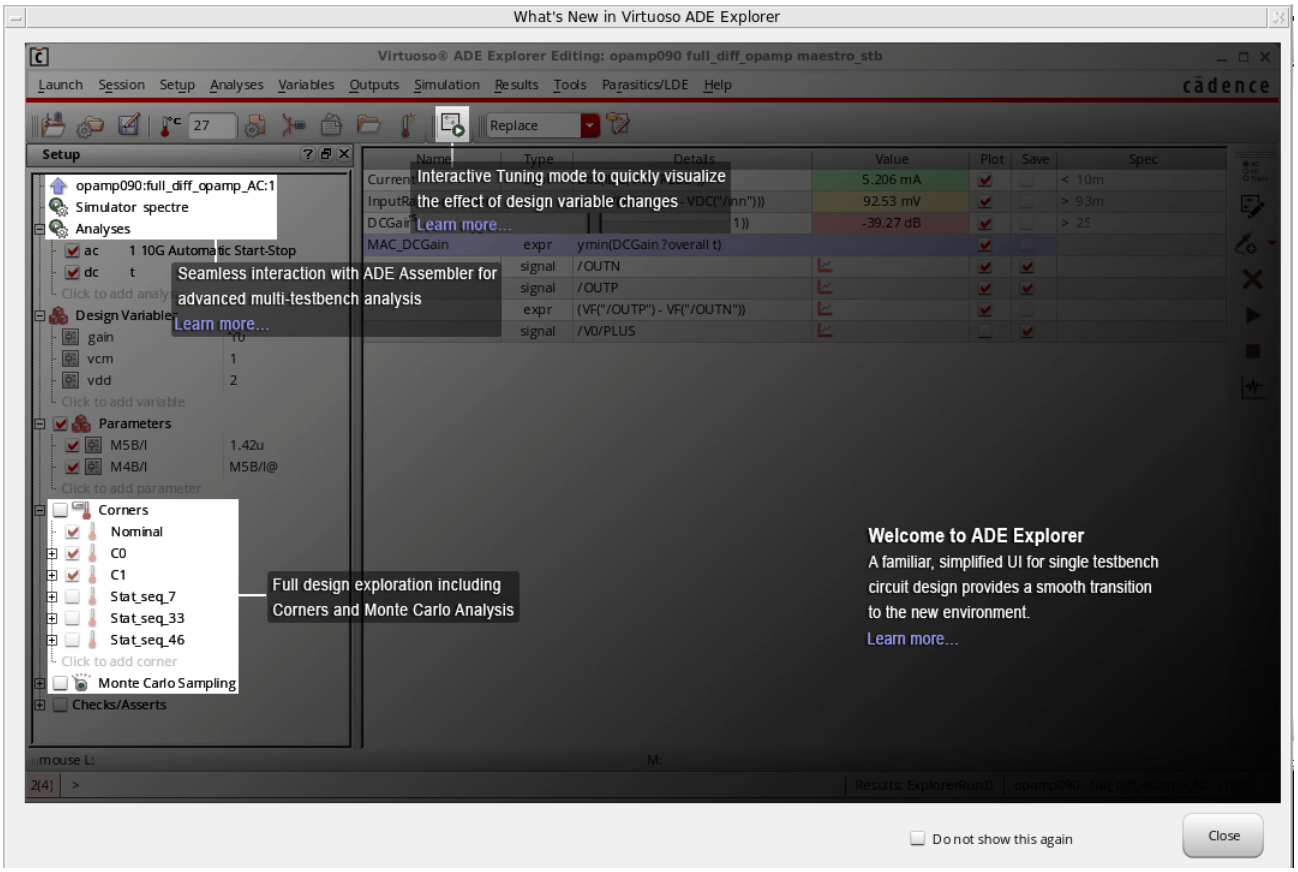
ADE Explorer is opened in the Edit mode. Design information is also displayed on the title bar





© 2023 Cadence Design Systems, Inc.

In addition, a *What's New in Virtuoso ADE Explorer* window appears when you open ADE Explorer for the first time in a given session.

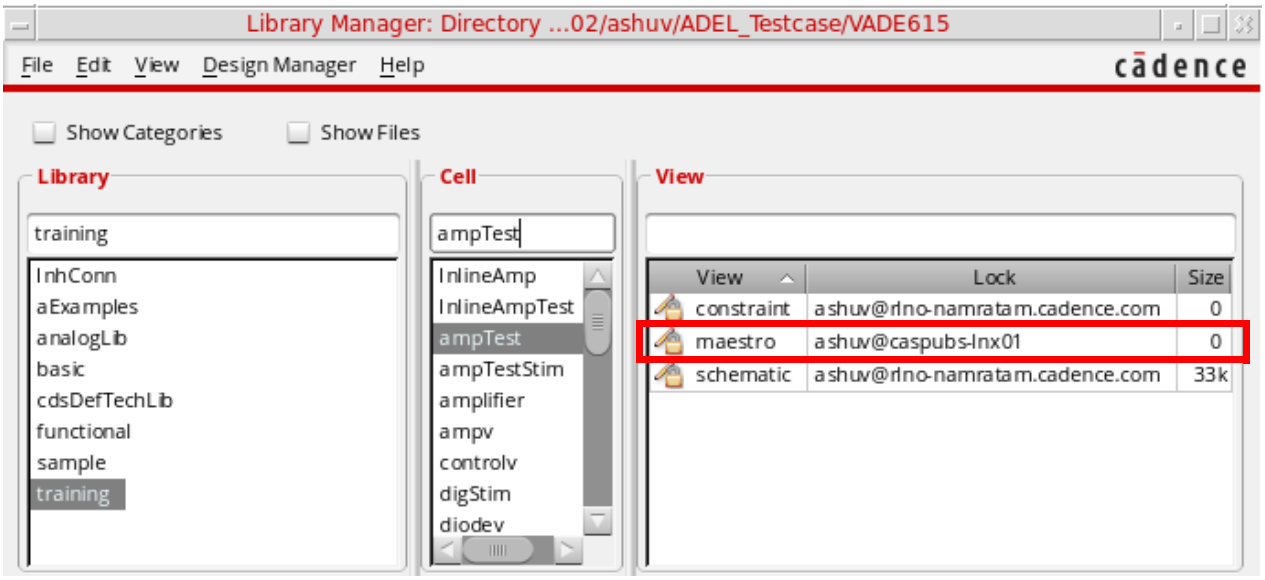



This window highlights the main features of ADE Explorer, such as real-time tuning mode, seamless interaction of ADE Explorer with ADE Assembler, and support for Monte Carlo analysis and Corners. You can click the *Learn More* hyperlinks provided within this window to know more about the specific features.

If you do not want to display this window next time when ADE Explorer is re-opened, select the *Do not show this again* check box displayed at the bottom. Click the *Close* button to close the window.


© 2023 Cadence Design Systems, Inc.

After the new maestro view is created, the *maestro* view is displayed in the *View* section of the Library Manager form in CIW:




You can save the current session settings by choosing *Session – Save* in ADE Explorer. The *maestro* view is saved. Next time, you can also open ADE Explorer through Library Manager by choosing the library, cell and double-clicking the *maestro* view.

**Note:** If you open an existing maestro view through the *File* menu in the CIW, it will open in the tool you worked in previously. For example, if you try to reopen a closed ADE Explorer session, the related maestro session will be opened in ADE Explorer again.

## Opening ADE Explorer From Virtuoso Schematic Editor

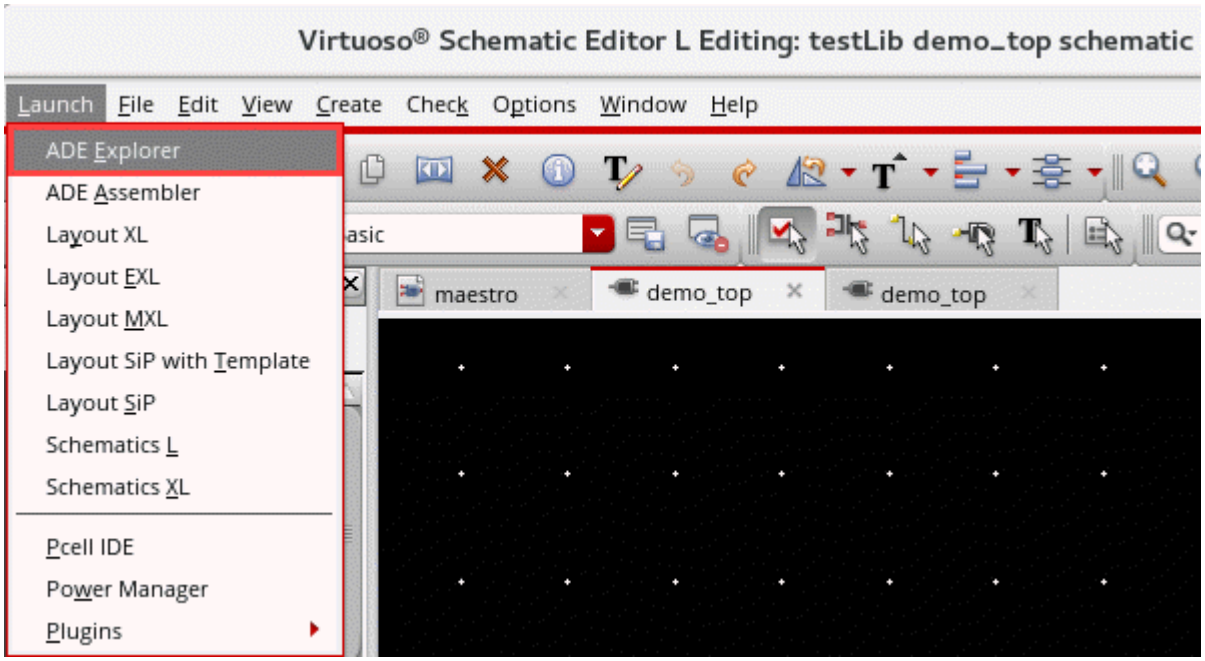
Follow the steps listed below to create a new maestro view and open Virtuoso ADE Explorer through the schematic window:

1. Open a design in the Virtuoso Schematic Editor window through the following CIW option:

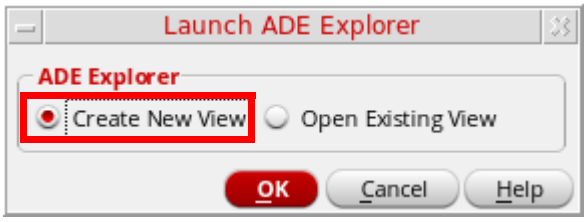
   - Choose *Tools – Library Manager*. The *Library Manager* form appears. Select a library, a cell, and double-click the schematic view.


© 2023 Cadence Design Systems, Inc.

The *Virtuoso Schematic Editor L* window appears as shown below.



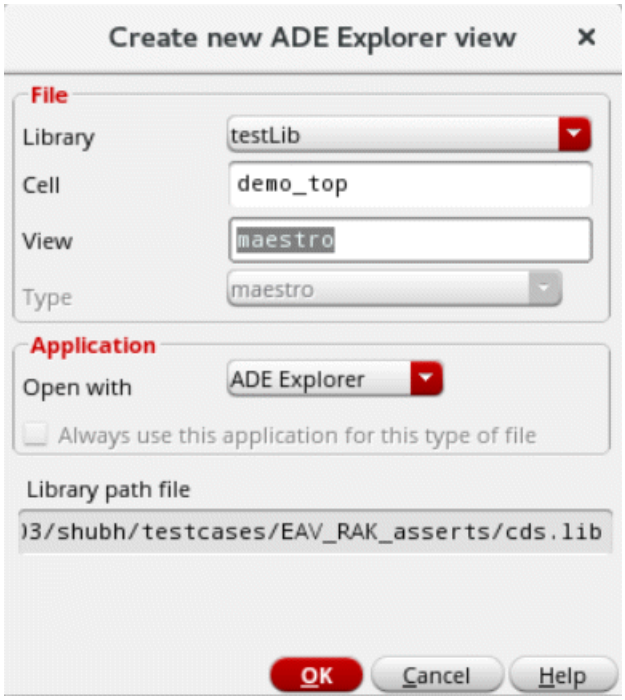

2. In the Schematic window, choose *Launch – ADE Explorer* to open Virtuoso ADE Explorer. The *Launch ADE Explorer* form appears as shown in the figure below.




3. In this form, select the *Create New View* option if you are opening ADE Explorer for the first time. If you already have a maestro (ADE Explorer) view set up, select *Open*


© 2023 Cadence Design Systems, Inc.

*Existing View*. When you choose to create a new view, the *Create new ADE Explorer view* form appears as shown below.



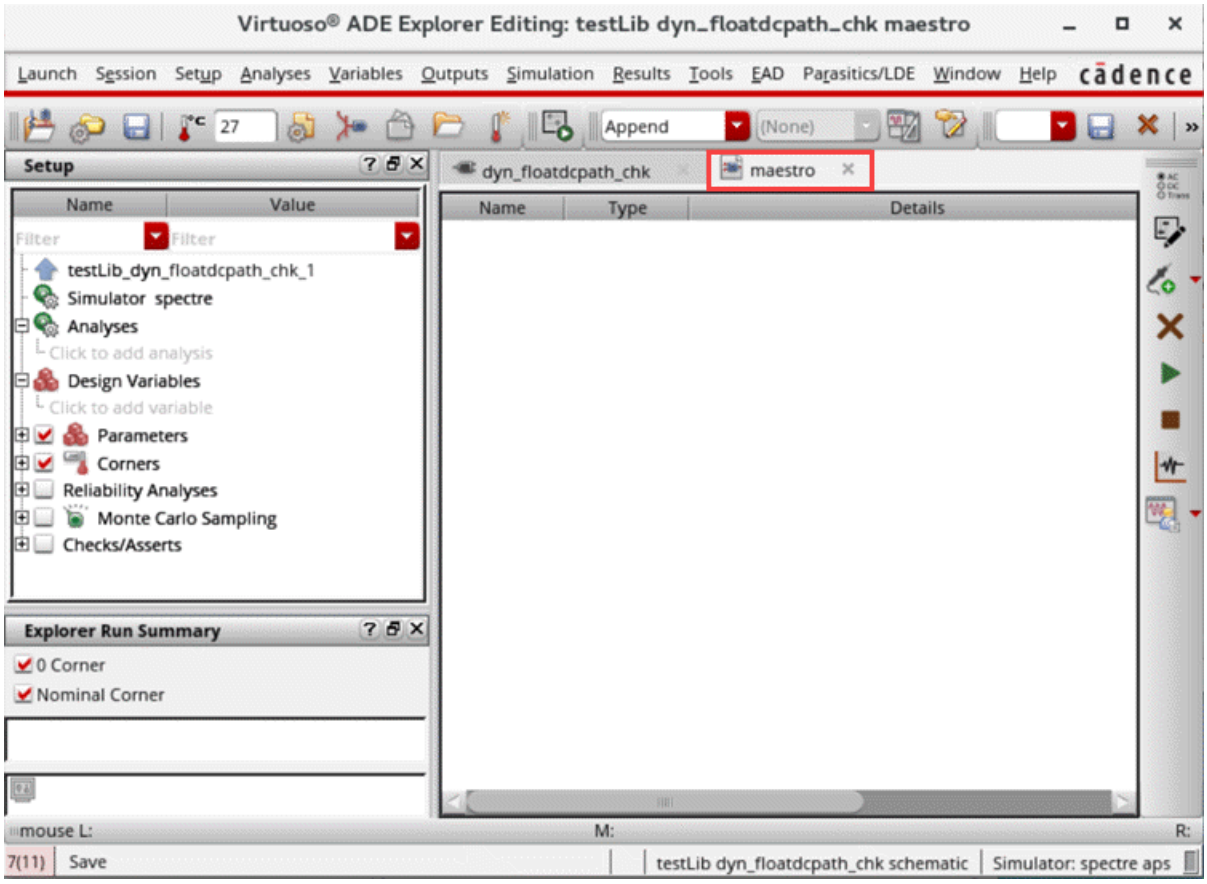

4. In this form, notice that the view type is already specified as `maestro` and the application type as `ADE Explorer`. You can specify whether you want to create a new tab in the Schematic Editor, replace the existing Schematic tab or create a new window to open ADE Explorer.

5. Click *OK*.

   The *Virtuoso ADE Explorer* window appears as shown below and the simulation environment is initialized for the design that is already open in Virtuoso Schematic Editor. You can now set up and run simulations for this design. Notice that a separate tab,


© 2023 Cadence Design Systems, Inc.

*maestro*, has been added for ADE Explorer indicating that the new tab option was selected at the time of creating the maestro cellview.




## Opening an Existing maestro View from Virtuoso Schematic Editor

To open an existing maestro view from Virtuoso Schematic Editor:

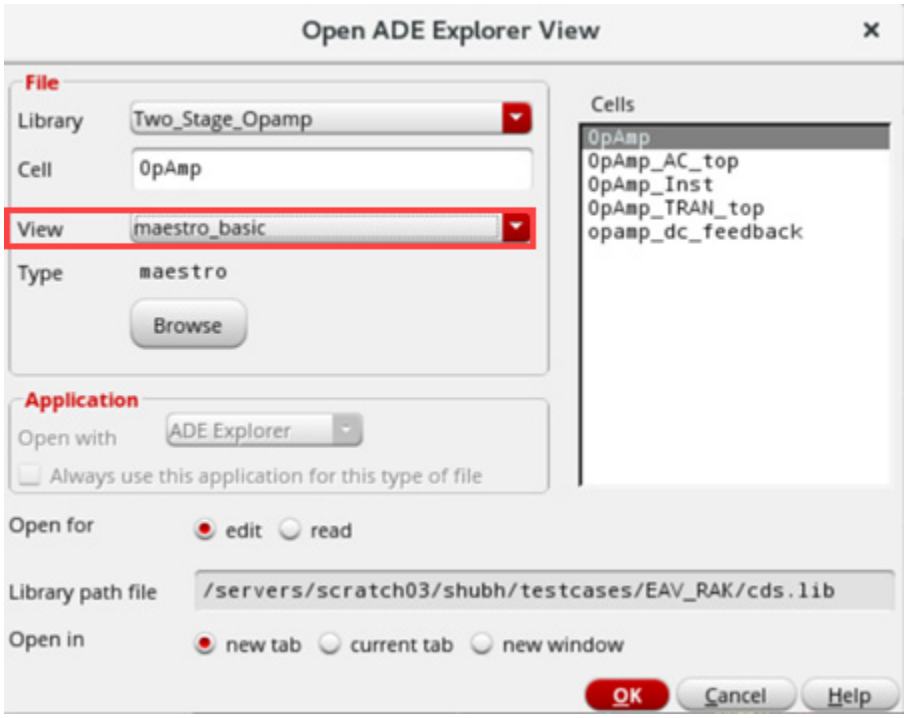
1. Choose *Launch – ADE Explorer*.

   The Launch ADE Explorer form is displayed.

2. In this form, select *Open Existing View*.


© 2023 Cadence Design Systems, Inc.

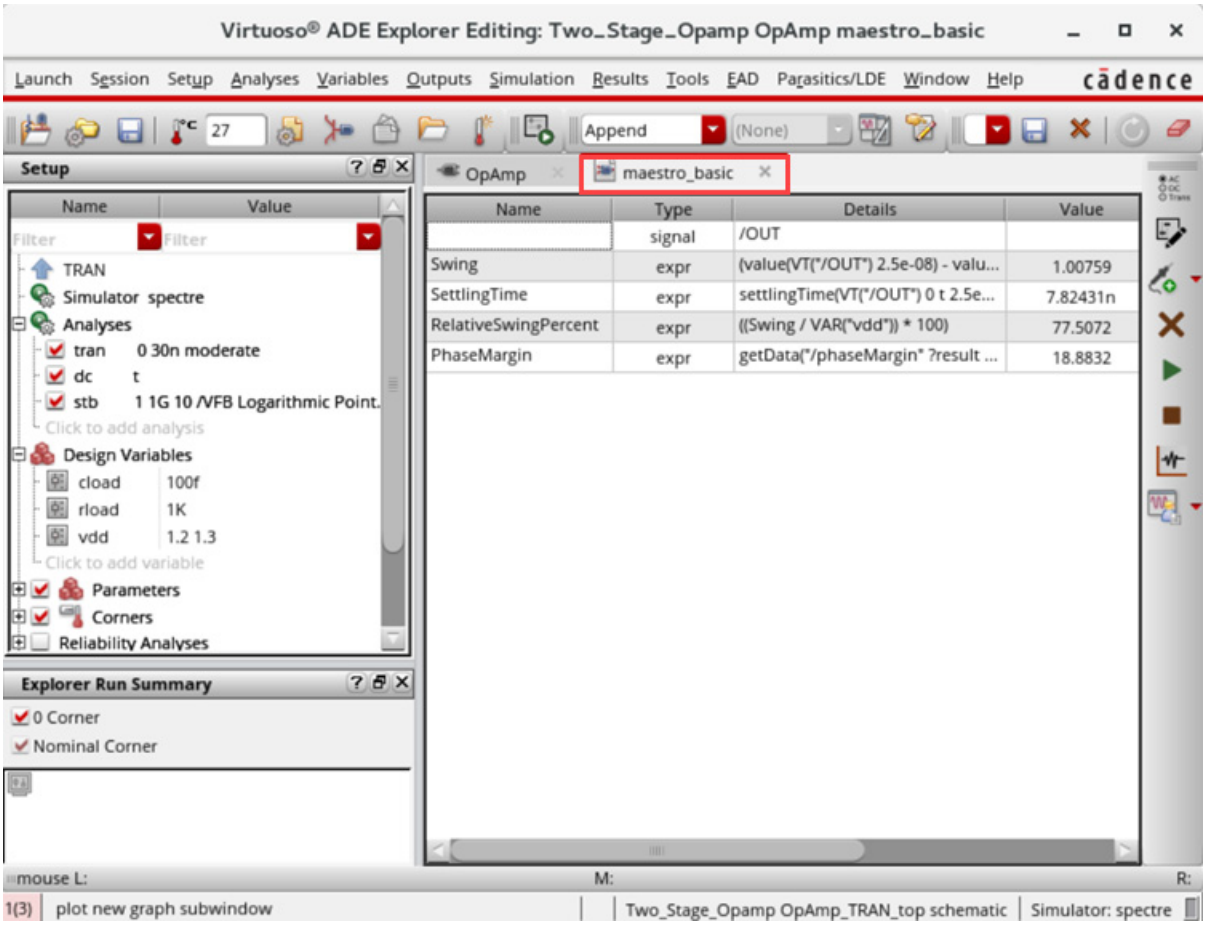
3. From the *View* drop-down list, select the maestro view you want to open.




4. Click *OK*.


© 2023 Cadence Design Systems, Inc.

The selected maestro view *maestro_basic* is opened.




## Default Design Edit Mode

You can set the `designEditMode` environment variable in CIW or `.cdsinit` file to specify the default mode in which your designs must be opened. The default value for this variable is `nil`, which means that the designs are opened in read-only mode by default. If you select `t`, the designs are opened in edit mode.

```
envSetVal("asimenv" "designEditMode" 'boolean t)
```

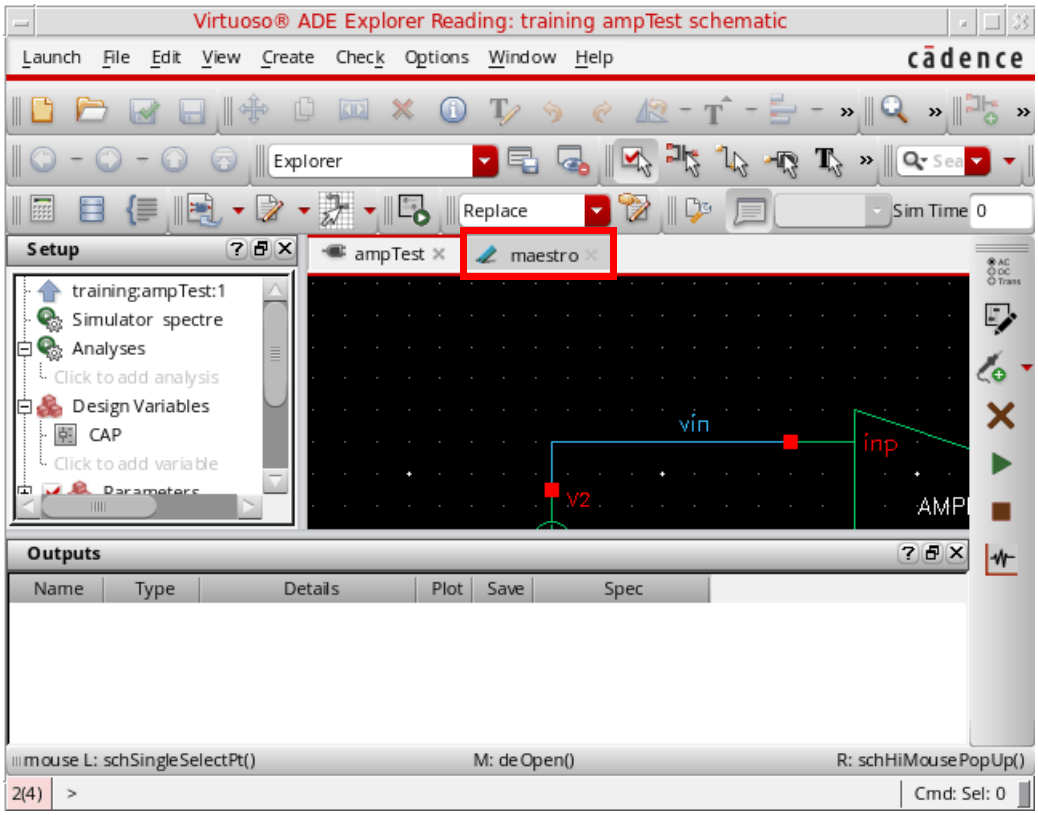
*Important*

When you open the `maestro` view from the Library Manager, it is opened in the same application from which it has been created. If it is created using ADE Explorer, it will open in ADE Explorer and if it is created using ADE Assembler, it will open in ADE Assembler.


© 2023 Cadence Design Systems, Inc.

## Opening ADE Explorer Using Schematic Editor Workspace

After you open the ADE Explorer maestro view, you can switch back to the schematic tab and change the workspace to `Explorer`. This option is provided in the Workspace Configuration toolbar of Virtuoso Schematic Editor.




This workspace displays the Setup and Outputs assistants that you can use to set up and run simulations and perform post-processing functions on the simulation results from the Schematic Editor window. This feature provides the following usability enhancements:

- You can now perform various simulation-related tasks without switching between the simulation and Schematic Editor window.
- If you change the simulation settings or customize menus from the simulation window, the Schematic Editor toolbars are also synchronized accordingly. For example, if you choose a simulator from the simulation window, the sub menu items in the Direct Plot menu change accordingly in both the simulation window menu and the Schematic Editor toolbar. Similarly, if you add a new sub menu item in the *Direct Plot* menu in the simulation window, the new sub menu item appears in the *Direct Plot* toolbar menu in the Schematic Editor.


© 2023 Cadence Design Systems, Inc.

## ADE Explorer Toolbars in Virtuoso Schematic Editor

The following table lists the ADE Explorer toolbar commands that are displayed along with the Virtuoso Schematic Editor toolbars when ADE Explorer is launched from schematic editor or vice versa.

| Command | Description |
|---|---|
| *Calculator* | Opens the Calculator window in Virtuoso Visualization and Analysis XL. |
| *Results Browser* | Opens the Results Browser window. |
| *Select Results* | Opens the Select Results form. |
| *Annotate* | Annotates and displays the selected outputs on schematic. |
| *Print* | Opens the schematic design where you can select the signals or instances that you want to print and then opens the Results Display Window to print the values. You can print the hard copies of selected components using this option. |
| *Direct Plot* | Opens the Direct Plot form that you can use to generate the quick waveform plots. |
| *Real Time Tuning* | Opens the Real Time Tuning assistant. |
| *Plotting Mode* | Lets you select the plotting mode. |
| *Plotting Template* | Lets you select the plotting template. |
| *Save plotting template* | Saves the plotting template. |
| *Plot Options* | Opens the Plotting Options form to specify plot options. |
| *Annotation Setup* | Opens the Annotation Setup form. |
| *Balloon Options* | Opens the Balloon Options form. |
| *Show/Hide Balloons* | Enables or disables info-balloons for waveforms or annotations. |
| *Clear Waveform Balloons* | Clears the pinned info-balloons. |
| *Choose analysis data to show in Waveform infoBalloons* | Lets you select the analysis for which the waveform info-balloons are to be displayed. |
| *Sim Time* | Lets you specify the transient time points. |


© 2023 Cadence Design Systems, Inc.

## ADE Explorer Menu in Schematic

When a schematic design is opened from within the ADE Explorer window, the *ADE Explorer* menu is displayed in the menu bar of the schematic window, as shown in the figure below.

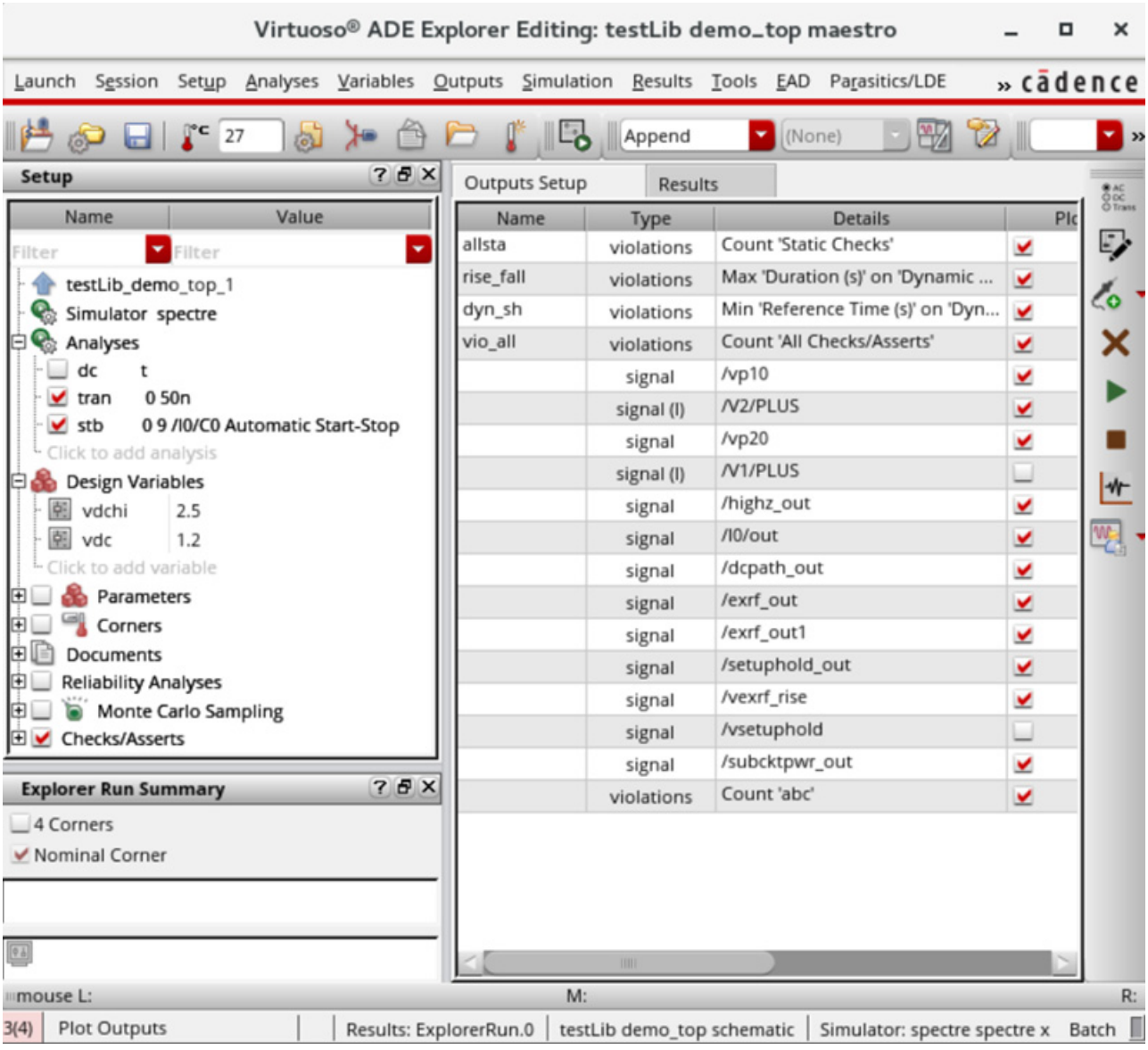
All the options in this menu are disabled when you work in the real-time tuning mode.


© 2023 Cadence Design Systems, Inc.

# Virtuoso ADE Explorer Graphical User Interface

The Virtuoso ADE Explorer user interface consists of a menubar, toolbars, sidebars, tabs, and assistants. You can customize the settings for these UI elements according to your requirements.

The title of the ADE Explorer window indicates whether the tool is opened in read or editing mode and also displays the library, cell, and view names.




Virtuoso ADE Explorer consists of the following components:


© 2023 Cadence Design Systems, Inc.

- Menu Bar
- Toolbars
- Assistants
- Outputs Setup Pane and Tabs
- Results Tab
- Status Bar

## Menu Bar

Menu bar consists of the following menus:

- Launch Menu
- Session Menu
- Setup Menu
- Analyses Menu
- Variables Menu
- Outputs Menu
- Simulation Menu
- Results Menu
- Tools Menu
- EAD Menu
- Parasitics/LDE Menu
- Help Menu


© 2023 Cadence Design Systems, Inc.

### Launch Menu

The table below lists the description of Launch menu:

| Menu Option Name | Description |
|---|---|
| *ADE Assembler* | Opens the design and simulation setup in Virtuoso ADE Assembler.<br><br>*Important*<br>You must have a valid Assembler license to open the setup in Assembler. |
| *Plugins – RealTime Tuning* | Opens the Real-Time Tuning assistant that you can use to change the variable and parameter values to fine-tune the design and view the updated plots simultaneously. |

### Session Menu

The table below lists the description of Session menu:

| Menu Option Name | Description |
|---|---|
| *Open Design in Tab* | Opens a new tab containing the design. |
| *Save* | Saves the current session settings for the library, cell, and view. |
| *Save a Copy* | Saves a copy of the current session settings for the library, cell and view. The Save a Copy form appears when you select this option. |
| *Import* | Opens the *Import Setup* form so you can import settings from an existing ADE Explorer setup database. |
| *Load ADE L State* | Loads the previously saved ADE L state in the current session. |
| *Save OCEAN Script* | Saves the current state as an OCEAN script. |
| *Quit* | Exits the ADE Explorer window. |


© 2023 Cadence Design Systems, Inc.

### Setup Menu

The table below lists the description of Setup menu:

| Menu Option Name | Description |
|---|---|
| *Job Setup* | Opens the *Job Policy Setup* form that you can use to set up a job policy to be used for the simulation run. |
| *Design* | Opens the *Choose Design -- ADE Explorer* form that you can use to choose another design (library, cell, and view) and opens the selected design in the current session in the read or edit mode. |
| *Simulator* | Opens the *Choosing Simulator -- ADE Explorer* form that you can use to choose a simulator. The default simulator is Spectre. The available simulators are—hspiceD, UltraSim, and ams |
| *Save Options* | Opens the *Save Options* form that you can use to specify simulation results directory and database location where the simulation results are saved. |
| *High-Performance Simulation* | Opens the *High-Performance Simulation Options* form that you can use to select the simulation performance mode. The available options are Spectre, APS, and XPS MS. |
| *Model Libraries* | Opens the *Model Library Setup* form that you can use to specify location and section information of the model files. |
| *Temperature* | Opens the *Setting Temperature* form that you can use to set the temperature. The default temperature is 27 degrees Celsius. |
| *Stimuli* | Sets the *Setup Stimuli* form that you can use to specify stimuli settings. |
| *Simulation Files* | Opens the *Simulation Files Setup* form that you can use to specify path and other settings for simulation files and vector files. The simulation files include—Definition files, Stimulus file, Parasitic files (DSPF and SPEF), and PSpice files. |
| *EM/IR Analysis* | Opens the *EM/IR Analysis Setup* form that you can use to specify settings for the Spectre EM/IR analysis. |


© 2023 Cadence Design Systems, Inc.

| Menu Option Name | Description |
|---|---|
| *MATLAB/Simulink* | ■ *Setting*—Opens the *Cosimulation Options* form that you can use to enables the co-simulation and specify its options.<br>■ *Log File*—Opens the log file. |
| *Environment* | Opens the *Environment Options* form that you can use to specify simulation environment options. |

### Analyses Menu

The table below lists the description of Analyses menu:

| Menu Option Name | Description |
|---|---|
| *Choose* | Opens the *Choosing Analysis* form that you can use to choose analyses for the specified simulator. When the simulator is set as Spectre, only `tran` analysis is selected by default. |
| *Delete* | Deletes the analysis selected in the Setup assistant. |
| *Enable* | Enables the analysis selected in the Setup assistant. |
| *Disable* | Disables the analysis selected in the Setup assistant. |

### Variables Menu

The table below lists the description of Variables menu:

| Menu Option Name | Description |
|---|---|
| *Edit* | Enables you to edit the already specified design variables. |
| *Delete* | Deletes the design variable selected in the Setup assistant. |
| *Find* | Opens the schematic and highlights the component on which the variable is applied. |


© 2023 Cadence Design Systems, Inc.

| Menu Option Name | Description |
|---|---|
| *Copy From Cellview* | Copies the selected design variables from a cellview. |
| *Copy To Cellview* | Copies the selected design variables to a cellview. |

### Outputs Menu

The table below lists the description of Outputs menu:

| Menu Option Name | Description |
|---|---|
| *Setup* | Opens the *Setting Outputs* form that you can use to specify output signals or expressions (created using ViVA) to be evaluated or plotted during the simulation run. |
| *Delete* | Deletes the selected output. |
| *Import* | Imports the outputs saved in a file. |
| *Export* | Exports the outputs to a file. |
| *Send To Expression Editor* | ■ *All*—Sends all the outputs to Expression Editor table in ViVA.<br>■ *Expressions*—Sends only the expressions to Expression Editor table in ViVA.<br>■ *Signals*—Sends only the signals to Expression Editor table in ViVA |


© 2023 Cadence Design Systems, Inc.

| Menu Option Name | Description |
|---|---|
| *To Be Saved* | Specifies the outputs you want to save during the simulation run.<br>■ *Select on Design*—Selects the outputs to be saved from the schematic design.<br>■ *Select OP Parameters*—Specifies the operating point parameters to be saved.<br>**Note:** Currently, this option is not available.<br>■ *Add To*—Adds the highlight to the selected set in the schematic and navigator. For example, if you select an output, `OUTP`, in the Outputs pane and click this option, the output is added to the selected set in the schematic and navigator.<br>■ *Remove From*—Removes the highlight from the selected set in the schematic and navigator. |
| *To Be Plotted* | Specifies the outputs you want to plot during the simulation run.<br>■ *Select on Design*—Selects the outputs to be saved from the schematic design.<br>■ *Add To*—Adds the highlight to the selected set in the schematic and navigator. For example, if you select an output, `OUTP`, in the Outputs pane and click this option, the output is added to the selected set in the schematic and navigator.<br>■ *Remove From*—Removes the highlight from the selected set in the schematic and navigator.<br>■ |
| *Save All* | Opens the *Save Options* form that you can use to specify save options, circuit information analysis options, and output format to be used for saving results. |


© 2023 Cadence Design Systems, Inc.

### Simulation Menu

The table below lists the description of Simulation menu:

| Menu Option Name | Description |
|---|---|
| *Netlist and Run* | Creates the netlist and runs the simulation. |
| *Stop* | Stops a running simulation. |
| *Run Preview* | Opens the *Run Preview* tab that shows the preview of the simulation setup summary. |
| *MDL Control* | Opens the *MDL Control Options* form that you can use to enable the MDL mode and specify the MDL mode options. |
| *Reliability* | ■ *Setup*—Opens the Reliability form that you can use to specify the reliability analysis settings.<br>■ *Save Options*—Saves the simulation results for reliability analysis.<br>■ *Load Options*—Loads the saved simulation results for reliability analysis |
| *Options* | ■ *Analog*—Opens the *Simulator Options* form that you can use to specify the options for the selected analog simulator.<br>■ *Digital*—Opens the *Simulator Options* form that you can use to specify the options for the selected digital simulator.<br>■ *Mixed Signal*—Opens the *Simulator Options* form that you can use to specify the options for the selected mixed-signal simulator. |
| *Netlist* | ■ *Create*—Generates the netlist for the current simulation run.<br>■ *Display*—Displays the netlist generated for the current simulation run.<br>■ *Recreate*—Recreate the netlist for the current simulation run. |
| *Output Log* | Opens the simulation log file. |


© 2023 Cadence Design Systems, Inc.

| Menu Option Name | Description |
|---|---|
| *Linter Log* | Displays the log file generated during lint checks. |
| *Convergence Aids* | *Node Set*—Sets an initial DC voltage on selected nodes<br>*Initial Condition*—Sets an initial transient voltage on selected nodes. |

### Results Menu

The table below lists the description of Results menu.

**Note:** When sweeps and/or corners are enabled in the simulation setup, the commands in the Results menu work on the root directory and generate the family outputs and/or plots.

| Menu Option Name | Description |
|---|---|
| *Plot Outputs* | ■ *Transient*—Plots the transient response for each node<br>■ *AC*—Plots the AC response for each node<br>■ *DC*—Plots the DC sweep response for each node<br>■ *Noise*—Plots the squared noise voltage for each node<br>■ *Expressions*—Plots the waveforms for expressions you define in the Setting Outputs form. |
| *Direct Plot* | *Main Form*—Opens the *Direct Plot* form that you can use to generate the quick waveform plots.<br>*Transient Signal*, *Transient Minus DC*, *Transient Sum*, *Transient Difference*, *AC Magnitude*, *AC dB10*, *AC dB20*, *AC Phase*, *AC Magnitude & Phase*, *AC Gain & Phase*, *Equivalent Output Noise*, *Equivalent Input Noise*, *Squared Output Noise*, *Squared Input* Noise, *Noise Figure*, *DC* |


© 2023 Cadence Design Systems, Inc.

| Menu Option Name | Description |
|---|---|
| *Print* | Opens the schematic design where you can select the signals or instances that you want to print and then opens the Results Display Window to print the values. You can print the hard copies of selected components using this option. |
| *Annotate* | Annotates and displays the selected outputs on schematic |
| *Vector* | ■ *Log File*<br>■ *Error File* |
| *Circuit Conditions* | Opens the *Circuit Conditions* form that you can use to perform the conditional search for results to find out components in the saturation region, breakdown region, or any user-defined region. |
| *Reliability Data* | You can select below options to view reliability results for:<br>■ *Device Lifetime and Degradation*<br>■ *Device Characteristic Degradation*<br>■ *Model Parameter Changes* |
| *EM/IR Data* | ■ *EM/IR Log*—Opens the log file for EM/IR analysis.<br>■ *Report*<br>■ *Layout Analysis*—Opens the selected layout view. |
| *Save* | Opens the *Save Results* form to save on-demand results (histories) in ADE Explorer that can be loaded later. This option is available only after you run the simulation. The name of the default history file is `Explorer.Run.0`. |
| *Select* | Opens the Select History form that displays the saved results and notes of a selected history item. |
| *Delete* | Deletes the saved results for a selected history. |
| *Plotting/Printing Options* | Opens the *ADE Explorer Plotting/Printing Options* form that you use to specify different plotting options. |


© 2023 Cadence Design Systems, Inc.

### Tools Menu

The table below lists the description of Tools menu:

| Menu Option Name | Description |
|---|---|
| *Corners* | Opens the *Corner Setup* form that you can use to specify the settings for corners to be created during the simulation run. |
| *Monte Carlo* | Opens the *Monte Carlo* form that you can use to setup the run options for Monte Carlo simulation. |
| *Export Data* | Opens the Export Results form that you can use to export the simulation results of a |
| *Create Datasheet* | Opens the Create Datasheet form that you can use to create datasheet for a checkpoint. |
| *Real Time Tuning* | Opens the Real Time Tuning assistant. |
| *RF* | ■ *ACPR*—Opens the ACPR Wizard.<br>■ *LSSP*—Opens the Large Signal S-Parameter Wizard. |
| *Calculator* | Opens the Virtuoso Visualization and Analysis XL Calculator. |
| *Results Browser* | Opens the Virtuoso Visualization and Analysis XL tool in Results Browser workspace. |
| *Waveform* | Opens the Virtuoso Visualization and Analysis XL Graph window. |
| *Results Display* | Opens the Results Display Window. |
| *Job Monitor* | Opens the Cadence Job Monitor window. |

### EAD Menu

The EAD menu helps to enable the Electrically Aware Design flow in Virtuoso and configure settings to save the electrical data.


© 2023 Cadence Design Systems, Inc.

**Note:** To be able to save the electrical data while running simulations, you require `Virtuoso_Layout_Suite_EAD` (95600) or `Virtuoso_Variation_Analysis_Op` (95510) license.

| Menu Option Name | 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.<br><br>For more information, see Selecting Signals to Save Currents in *Virtuoso Electrically Aware Design Flow Guide*. |

**Note:** For details on how to use simulation results in the EAD flow, see Running Simulations and Viewing Output Current Data in *Virtuoso Electrically Aware Design Flow Guide*.

### Parasitics/LDE Menu

**Note:** This menu is not available when you work in the real-time tuning (RTT) mode.

| Menu Option Name | Description |
|---|---|
| *Setup* | Opens the Parasitics & LDE Setup form that you can use to set up the parasitic information about your design. |
| *Create Estimates* | Ceates the estimated parasitics and builds a parasitic or an LDE view. |
| *Create Filters* | Creates the parasitic filters to refine the extracted view. |
| *Report* | Creates a range of parasitic reports (including comparing the parasitic/LDE and extracted views), exports the report information, and cross-probes with the extracted view. |
| *Compare* | Helps to setup the parasitic comparisons for the parasitic/LDE schematic and extracted views. |


© 2023 Cadence Design Systems, Inc.

| Menu Option Name | Description |
|---|---|
| *Options* | Opens the Options form that you can use to set the options related to backannotation and parasitic probing for estimated and extracted parasitics. |

**Note:** This menu is not available in the real-time tuning (RTT) mode.

For more information about these options, see *Virtuoso Parasitic Aware Design User Guide*.

### Window Menu

The table below lists the description of Windows menu:

| Menu Option Name | Description |
|---|---|
| *Assistants* | Lists the following assistants:<br>■ *Parasitics & Electrical Setup*<br>■ *Parasitics Filters*<br>■ *Parasitics Report*<br>■ *Explorer Run Summary*<br>■ *Setup*<br>■ *Operating Region*<br>■ *Variables and Parameters*<br>T*oggle Visibility*—Changes the show or hide status of the selected assistants. |
| *Toolbars* | Lists the available toolbars. For information about the toolbars list, see Toolbars on page 68.<br>*Toggle Visibility*—Changes the show or hide status of the selected toolbars.<br>*Customize*—Opens the Toolbar Manager that you can use to customize toolbars. |


© 2023 Cadence Design Systems, Inc.

| Menu Option Name | Description |
|---|---|
| *Workspaces* | Sets the workspace. The available workspaces are: *Basic* and *Real Time Tuning*.<br>This option also lists the commands that you can use to perform workspace-specific tasks:<br>■ *Save As—Opens the Save Workspace form that you can use Saves the Workspace*<br>■ *Delete*<br>■ *Load*<br>■ *Set Default*<br>■ *Revert to Saved* |
| *Tabs* | List the tabs that are currently open in ADE Explorer. To open a tab, click the tab name.<br>■ *Close Current Tab*—Closes the currently-active tab.<br>■ *Close Other Tabs*—Closes all the additional tabs. Only ADE Explorer window remains open. |

### Help Menu

The table below lists the description of Help menu:

| Menu Option Name | Description |
|---|---|
| *Search* | A text field that lets you enter a search string. Press `Enter` to view the search results.<br>**Note:** Do not enclose the search string in double quotes. |
| *User Guide* | Opens Virtuoso ADE Explorer User Guide (at the section that provides information about using CIW) in Cadence Help. |
| *What's New* | Opens the Virtuoso ADE Explorer What's New document in Cadence Help. |


© 2023 Cadence Design Systems, Inc.

| Menu Option Name | Description |
|---|---|
| *Known Problems and Solutions* | *Opens the Virtuoso Known Problems and Solutions document* in Cadence Help. |
| *Virtuoso Documentation Library* | Opens the Cadence Help home page, which provides quick access links to the following local and online resources:<br>■ What's New<br>■ Video Demos and Tutorials<br>■ Featured Content<br>■ Known Problems and Solutions<br>■ Other web resources |
| *Virtuoso Video Library* | Opens the Video Library page available on Cadence Online Support (COS). This page lists the videos available for various Virtuoso products.<br>**Note:** You must have a COS account to access the content available on COS.<br>**Note:** Contact your IT support to ensure that the Internet ports required for video playback are enabled. |
| *Virtuoso Rapid Adoption Kits* | Opens the Rapid Adoption Kits page on COS. This page lists Rapid Adoption Kits (RAKs) available for various Virtuoso products. |
| *Virtuoso Learning Map* | Lists domain-specific training available on Cadence Training Services.<br>Cadence Training Services learning maps provide a comprehensive visual overview of the learning opportunities for Cadence customers. They provide recommended course flows as well as tool experience and knowledge levels to guide customers through a complete learning plan. |


© 2023 Cadence Design Systems, Inc.

| Menu Option Name | Description |
|---|---|
| *Virtuoso Custom IC Community* | Opens the Virtuoso Custom IC Community web page. This page provides access to the latest blogs and discussion threads on various Virtuoso products and design topics, information about software downloads and support and training, and other related information. You too can contribute to the community forum by creating a Cadence account. This gives you additional benefits such as alerts about topics of interest and access to online webinars. |
| *Cadence Online Support* | Opens COS, which you can use to access information about Cadence products, documentation, videos, RAKs, application notes, troubleshooting information, alerts, and so on. Improvements are regularly made to COS to make it convenient for you to look up the information you want. We recommend that you bookmark this web site and use it as your first point of reference for any Virtuoso-related information.<br><br> *Tip*<br>You can also access COS by clicking the Cadence logo available in the upper-right banner in each Virtuoso window.<br><br> |
| *Cadence Training* | Opens the Cadence training web page. You can find on this page information about the training courses available in different regions. Information is available about both classroom and online courses. |
| *Cadence Community* | Opens the Cadence Community web page. This page provides access to the latest blogs and discussion threads on various Cadence products and solutions, and EDA Industry Insights. You too can contribute to the community forum by creating a Cadence account. This gives you additional benefits such as alerts about topics of interest and access to online webinars. |
| *Cadence OS Platform Support* | Provides information about the current Cadence software releases and the supported platforms. |


© 2023 Cadence Design Systems, Inc.

| Menu Option Name | Description |
|---|---|
| *Contact Us* | Opens the Cadence Customer Support web page, which provides customer support contact information for different regions. |
| *Cadence Home* | Opens the Cadence corporate web site. |
| *About Virtuoso* | Displays Virtuoso Studio Design Environment version information. |


© 2023 Cadence Design Systems, Inc.

## Toolbars

To show or hide toolbars in Virtuoso ADE Explorer:

➡ Right-click anywhere in the menu bar and select the toolbars that you want to show.

The Virtuoso ADE Explorer consists of the following toolbars:

- Run Toolbar
- Explorer Simulation Setup
- Real Time Tuning Toolbar
- Results Toolbar
- Plotting Toolbar
- Output Setup Toolbar
- Parasitic Mode Toolbar
- Go Toolbar
- Bookmarks Toolbar
- Workspaces Toolbar

### Run Toolbar

The Run toolbar contains the following buttons:

- *Choose Analysis*— Opens the Choosing Analyses form.


© 2023 Cadence Design Systems, Inc.

- *Edit Variables*— Opens the Editing Design Variables form.
- *Add Outputs*—Includes a drop-down to specify the output you want to add.
- *Delete*— Deletes the selected outputs from the *Output Setup* tab.
- *Run Simulation*— Runs a simulation.
- *Stop Simulation*— Stops a simulation.
- *Plot Outputs*— Plots all available results at once.

### Explorer Simulation Setup

The ADE Explorer Simulation Setup toolbar contains the following buttons:

- *Choose Design*
- *Choose Simulator*
- *Save*
- *Temperature*
- *Model Libraries*
- *Stimuli*
- *Environment*
- *Load ADE L State*
- *Corner Setup*

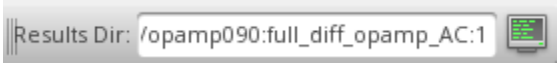
### Real Time Tuning Toolbar

The Real Time Tuning toolbar contains the following button that opens the Real Time Tuning assistant to the left of the ADE Explorer window:

© 2023 Cadence Design Systems, Inc.

### Results Toolbar

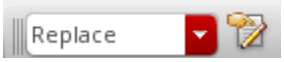
The Results toolbar contains the following buttons:




- *Results Directory*—Specifies the simulation results directory location.
- *Xterm*—Opens the terminal window.

### Plotting Toolbar

The Plotting toolbar contains the following buttons:




- *Plotting Mode*—Specifies the plotting modes that you want to use for plotting output waveforms. The available modes are—Append, Replace, New Subwindow, and New Window
- *Plot Options*—Opens the Plotting Options form to specify plot options.

### Output Setup Toolbar

The *Output Setup* toolbar is displayed by default and is used to create and save named filters for combination of column searches that you use or refer to more frequently. These named filter include search words or strings used to filter the outputs for all columns.

- *Named filter* drop-down list—Lists the named filters.
- *Save named filter*—Saves the specified named filter.
- *Delete named filter*—Deletes the specified named filter.
- *Reset Table*—Restores the original arrangement or undoes any sorting of the output rows.
- *Remove .tmpADEDir*—Deletes the `.tmpADEDir` directory.

  The `.tmpADEDir` directory contains the intermediate simulation and netlisting information for a given mestro cellview. The directory can be used in parallel in a different Virtuoso session for the same testbench. It is recommended to remove the `.tmpADEDir`


© 2023 Cadence Design Systems, Inc.

carefully because though it can help in resolving netlist issues, cleaning the `.tmpADEDir` directory increases the run time for the incremental simulations. It removes data for all the tests in the cellview. If this command is used in ADE Explorer, the .tmp data is removed for all tests, including data that may be part of other tests in ADE Assembler.

### Parasitic Mode Toolbar

The Parasitic Mode toolbar is not displayed by default and appears after the parasitics have been setup using the Parasitics/LDE Setup form. This toolbar contains the following buttons:

- Parasitic/LDE Mode—Specifies the parasitics mode from the drop-down list. The default mode is `No Parasitics/LDE`. The other available modes are: `Schematic Estimates (Parasitics)`, `Layout (Parasitics/LDE)`, and `Extracted (Parasitics/LDE)`.
- Setup (Parasitics/LDE)—Opens the Parasitics/LDE Setup form that you can use to set up parasitics.
- Build Parasitics/LDE view (Custom Parasitics)—Creates the custom parasitics or LDE view.
- Sweep Drop-down list – Specifies whether to sweep the devices or parasitics. The available options are: `No Sweeps`, `Devices Only`, `Parasitics Only`, and `Devices and Parasitics`.

  For more information about these options, see *Virtuoso Parasitic Aware Design User Guide*.

**Note:** This toolbar is not available in the real-time tuning (RTT) mode.

### Go Toolbar

The Go toolbar contains the following buttons:

- Back
- Forward
- Up


© 2023 Cadence Design Systems, Inc.

- Top

### Bookmarks Toolbar

This toolbar contains the list of bookmarked views.

### Workspaces Toolbar

The Workspaces toolbar contains the following buttons:

-

- Workspace drop-down—Displays the list of available workspaces—Basic and Real Time Tuning.
- Save Workspace—Opens the Save Workspace form to create and save a new workspace.
- Toggle Assistants—Controls the display of assistants in ADE Explorer.


© 2023 Cadence Design Systems, Inc.

## Assistants

To show or hide assistants in Virtuoso ADE Explorer:

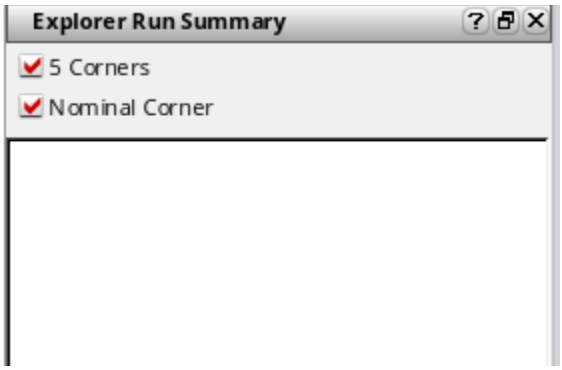
➡ Right-click anywhere in the menu bar and select the assistants that you want to show.

Virtuoso ADE Explorer consists of the following assistants:

- Explorer Run Summary Assistant
- Variable and Parameter Assistant
- Setup Assistant
- Real Time Tuning Assistant
- ViVA Graph Assistant

### Explorer Run Summary Assistant

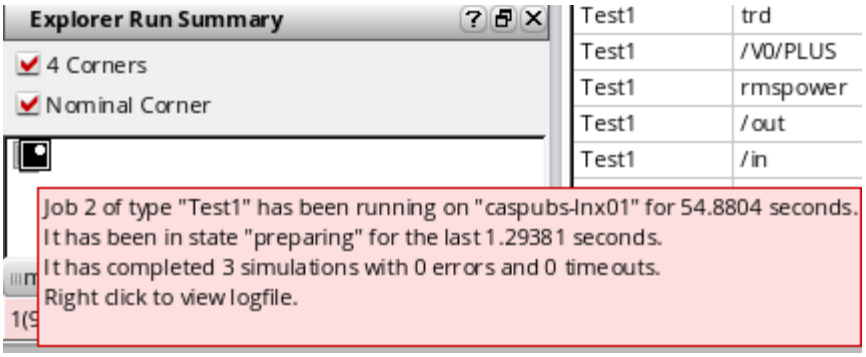
The Explorer Run Summary assistant displays the number of corners running during a simulation run.



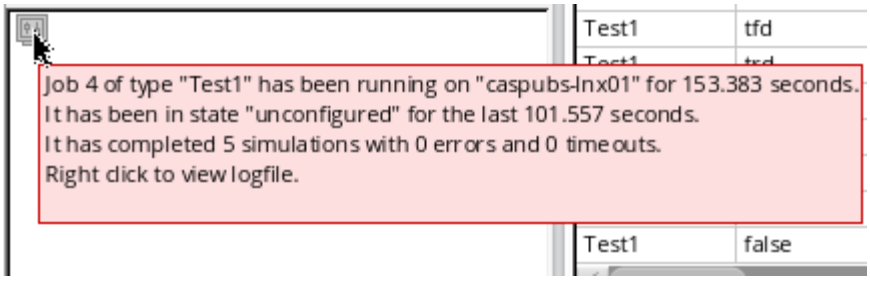


© 2023 Cadence Design Systems, Inc.

During the simulation run, a progress symbol ( ) appears on the Explorer Run Summary assistant. When place your mouse over this symbol, the simulation run status appears in a tooltip, as shown in the figure below:




When the simulation run is complete, it shows the completion status, as shown in the figure below:



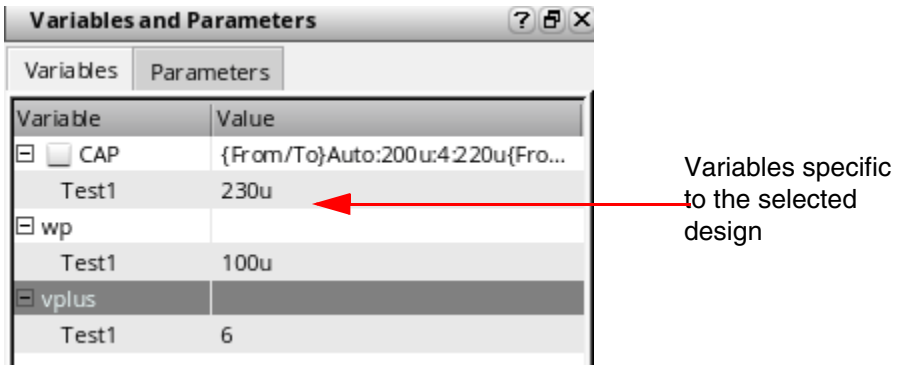

### Variable and Parameter Assistant

The Variable and Parameters assistant includes two tabs—*Variables* to create and view the global variables and variables local to the selected design, and *Parameters* to create and view parameters.

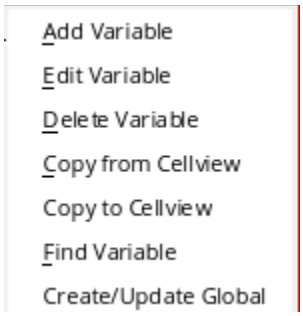

© 2023 Cadence Design Systems, Inc.

### *Variables Tab*

Lists the design variables displayed in the Setup assistant and their values.



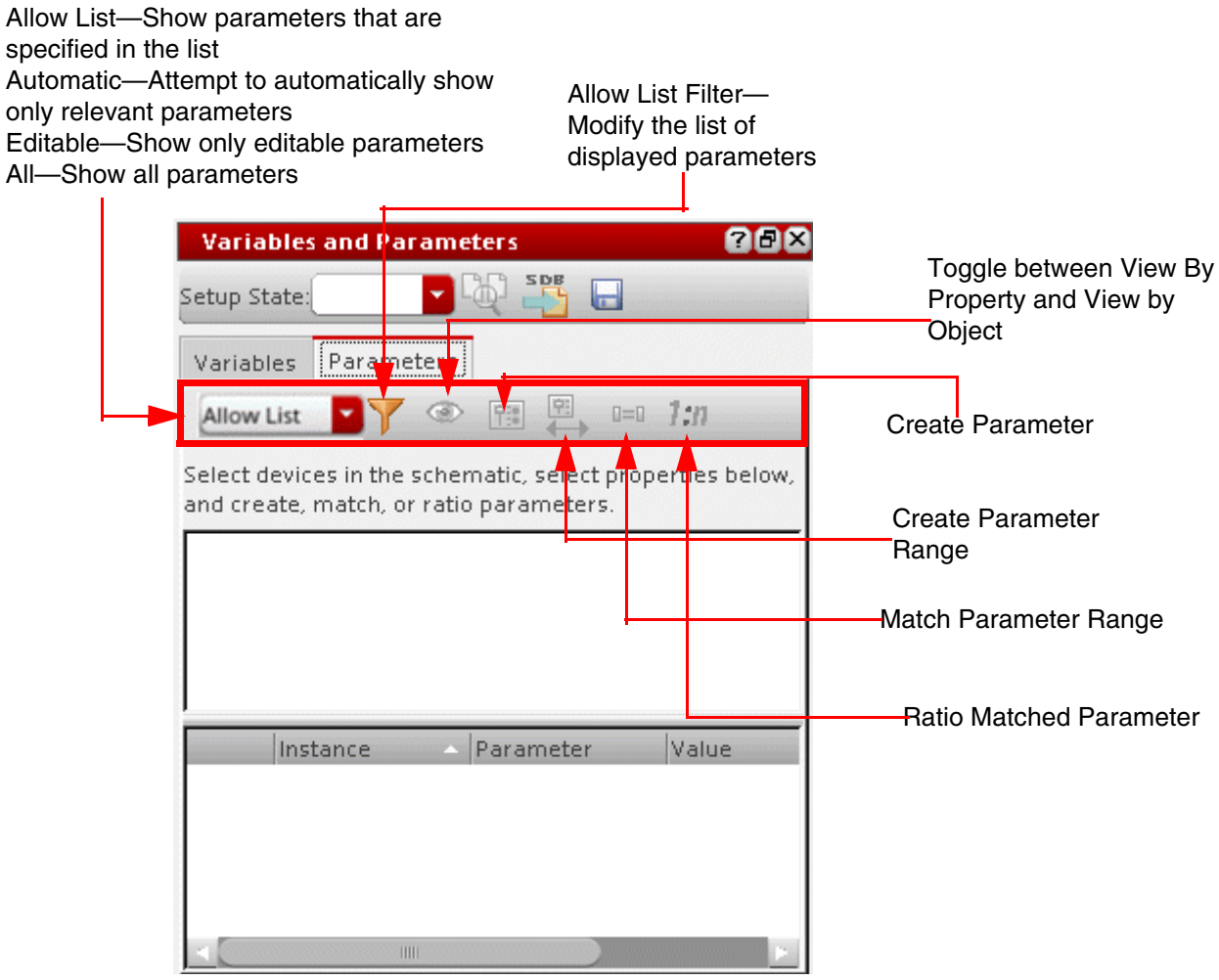

When you right-click the test name or the variable value, the following context menu appears that you can use to add, edit, delete, copy, and find variables:





© 2023 Cadence Design Systems, Inc.

## *Parameters Tab*

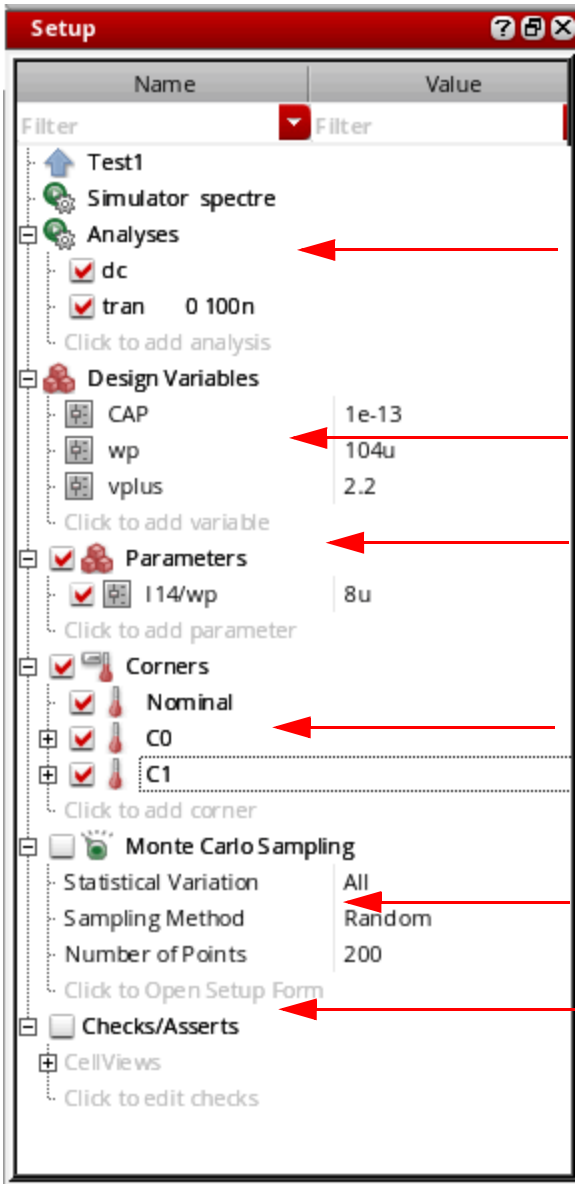




© 2023 Cadence Design Systems, Inc.

## Setup Assistant

The Setup assistant helps you specify and view the settings for analyses, design variables, parameters, corners, Monte Carlo simulation run options, and checks/Asserts.




Choose Analyses form appears that you can use to add analyses

Editing Design Variables Form appears that you can use to add variables

Schematic design appears with Parameters and Variables that you can use to add parameters and variables.

Corner Setup form appears that you can use to add corners

Monte Carlo setup form appears that you can use to set Monte Carlo run options

Checks and Asserts assistant appears in the schematic view

The Setup assistant includes the following two columns to display the setup data:

- *Name*—Includes the names of the setup elements, such as variable, parameter, corner, and so on.
- *Value*—Includes the values of the elements defined in the setup.


© 2023 Cadence Design Systems, Inc.

These columns are separated by a divider bar that you can drag to change the width of the columns. The width of the divider bar can also be controlled by the nameDisplayWidthInDataView `.cdsenv` variable.

### *Filtering Data in the Setup Assistant*

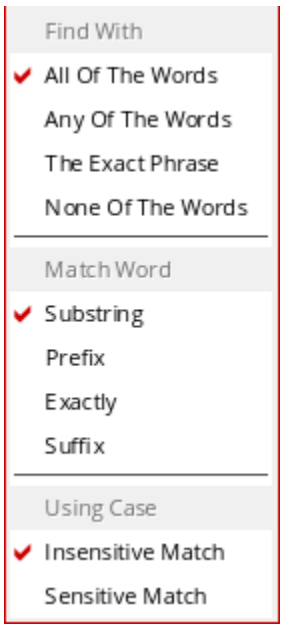
The Setup assistant includes search filters for the *Name* and *Value* columns. You can use these filters to filter the data based on the specified criteria.

These filters are displayed by default. To show or hide filters,

- Right-click anywhere in the column header and choose *Hide Filters* or *Show Filters*.
- Set the showFilterInDataView `.cdsenv` variable.

To specify filter conditions while searching or filtering the outputs,

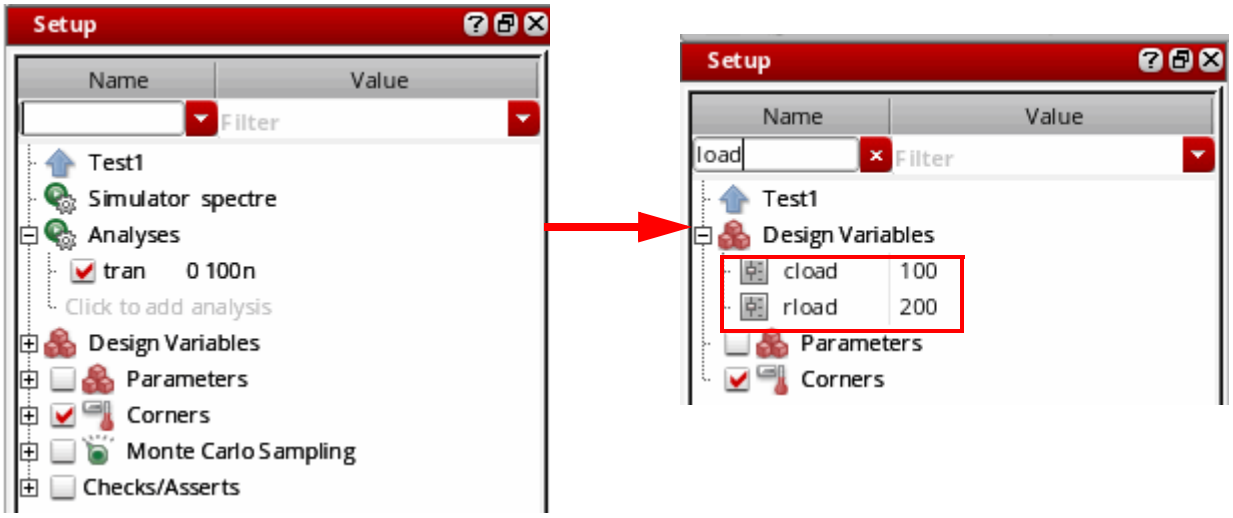
- Right-click the *Filter* drop-down list and select the conditions listed in the figure below. This figure displays the conditions that are selected by default.





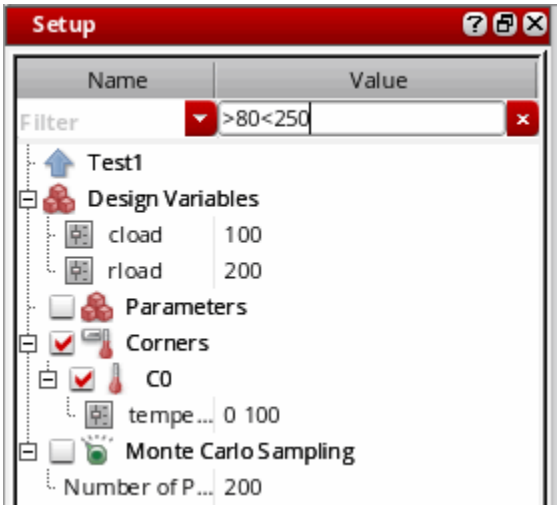
© 2023 Cadence Design Systems, Inc.

When you type a search string in the *Filter* drop-down in the *Name* or *Value* column, the words matching the search criteria are displayed. In addition, the sections are expanded to show the data that matches with the specified search string. For example, the following figure shows the *Design Variables* section expanded when you search the string `load`.




When you clear the search string, the Setup assistant is returned back to its original non-expanded form.

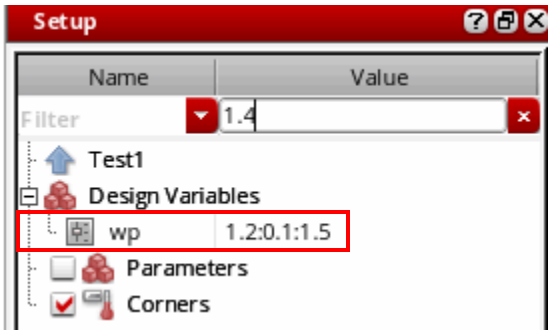
Similarly, the *Value* column displays the words matching the search string. You can also search for a range of values in the *Value* column, and the values that fall in the specified range are displayed, as shown in the figure below.





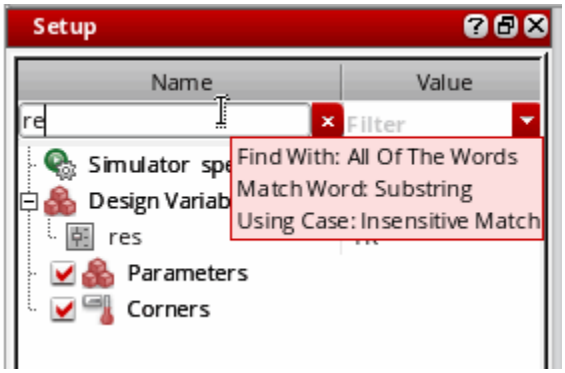
© 2023 Cadence Design Systems, Inc.

You can also search for a value that is a part of the sweep range specified in the variables and parameters. For example, if the variable `wp` includes sweep range `1.2:0.1:1.5`, and you search for `1.4`, then the following results are displayed.




**Note:** The search filters are retained when you transition from ADE Explorer to ADE Assembler and ADE Assembler to ADE Explorer.

If you move the mouse over the *Filter* drop-down, the filter conditions are displayed.




A history of the searched words or strings is also maintained in the *Filter* drop-down list. This history can be reused later, and therefore, helps in making the filtering or searching faster. In addition, you do not need to type the text again to search for a particular output.

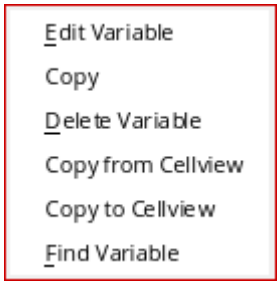
To clear the search history, select *Clear Search History* from the drop-down list of the search filter.

In addition, after the searched words are filtered and listed, the *Filter* drop-down list changes into a *Cancel* button that you can use to cancel the search and display all the specified output.


© 2023 Cadence Design Systems, Inc.

The following context-menu appears when you right-click the design information:

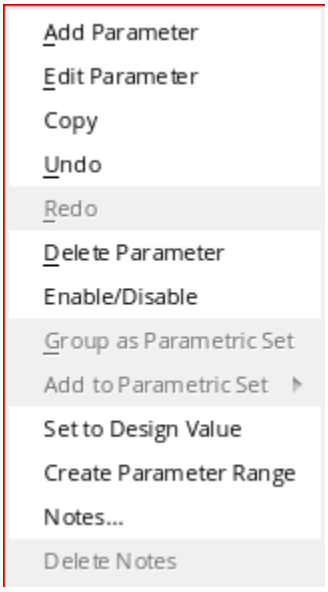
The following context-menu options appear when you right-click a design variable:



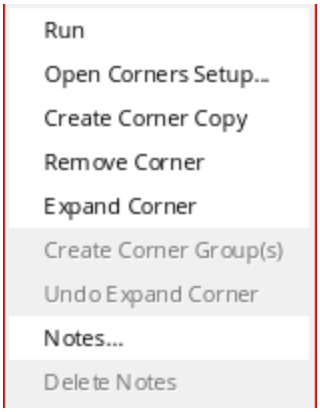


© 2023 Cadence Design Systems, Inc.

The following context-menu options appear when you right-click a parameter:




The following context-menu options appear when you right-click a corner:




### Real Time Tuning Assistant

The Real-Time Tuning (RTT) assistant in ADE Explorer helps you edit the values of variables and parameters and the simulation results are dynamically updated to reflect the changes that have been made.

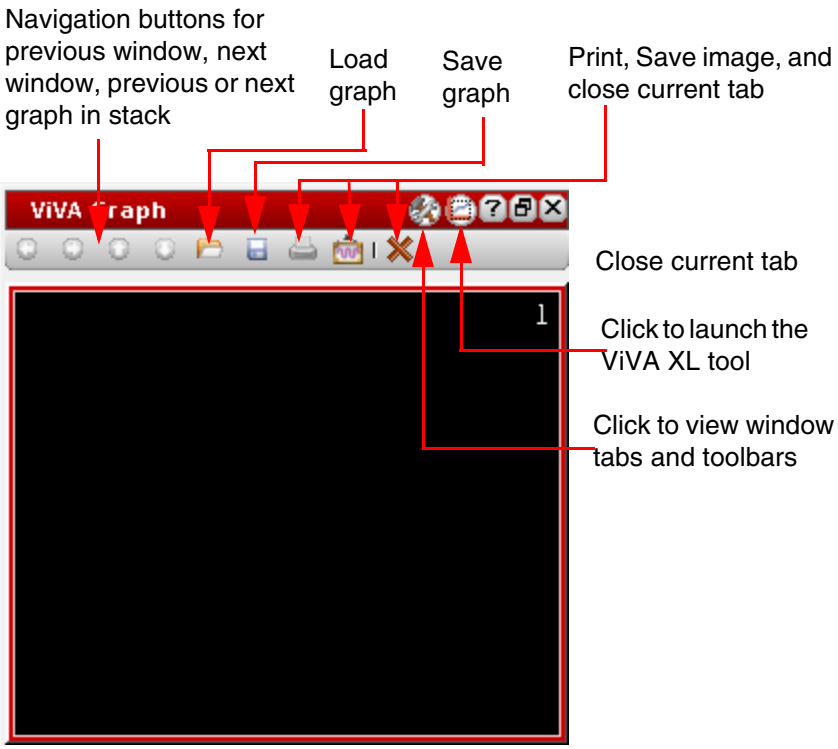
For more information about this assistant, see Chapter 10, “Real-Time Tuning in Simulations”.


© 2023 Cadence Design Systems, Inc.

### ViVA Graph Assistant

The Virtuoso Visualization and Analysis XL graph displays the output waveforms after a simulation run is complete. If the  environment variable is set to `Assistant`, the Virtuoso Visualization and Analysis XL graph appears as an assistant in ADE Explorer and displayed to the right of the ADE Explorer or design window.

By default, this assistant does not show separate tabs that are created when you select plotting mode as `New Win`. To view the tabs, right-click anywhere on the toolbar and choose *Show Tab Bar*. The tabs are displayed below the toolbar.




In ADE Explorer, you can open this assistant through one of the following ways:

- Right-click anywhere in the menu bar and choose *Assistants – Graph*.
- Choose *Tools – Waveform*.
- Click the button on the Plotting toolbar.

In Schematic Editor, you can open the graph assistant through one of the following ways:

- Right-click anywhere in the menu bar and choose *Assistants – Graph*.
- Choose *Windows – Assistants – ViVA Graph*.


© 2023 Cadence Design Systems, Inc.

- Click the button on the Plotting toolbar.

  If the  environment variable is set to `Assistant`, the Virtuoso Visualization and Analysis XL graph appears as an assistant.

When the Virtuoso Visualization and Analysis XL graph assistant is opened for the first time in a session, the following pop-up appears that shows how you can undock and re-dock the graph assistant in ADE Explorer.

You can select the `Do not show this again` check box to disable the dialog box. You can also set the following environment variable in CIW and .cdsinit file to control the display of this pop-up.

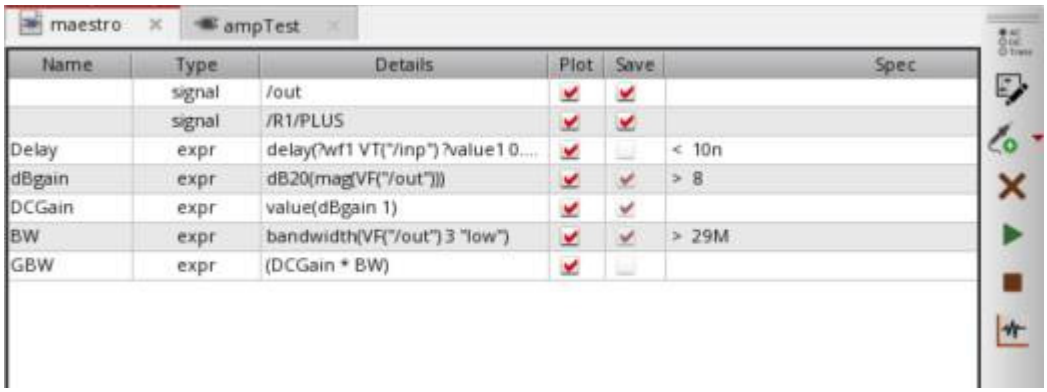
```
envSetVal("maestro.plotting" "showViVAGraphLaunchTip" 'boolean t)
```

**Note:** When the subwindows are already open in the graph assistant and you change the assistant position by docking it at the bottom of schematic, the graph layout automatically changes to horizontal. Similarly, docking assistant along the sides changes the graph layout to vertical.

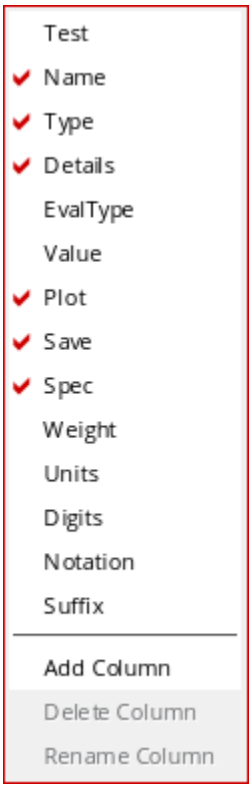

© 2023 Cadence Design Systems, Inc.

## Outputs Setup Pane and Tabs

Outputs Setup pane is a static component that displays the outputs that you want to be saved, evaluated, and plotted during the simulation run. Tabs are also displayed above this pane that you can use to switch between the ADE Explorer and Schematic environments.




The header of the Outputs Setup pane contains columns to display specific details for each output, such as output name, type, details, value, whether to be saved or plotted, spec, and so on. To display or hide these columns, right-click anywhere in the header and choose the column name you want to show or hide.



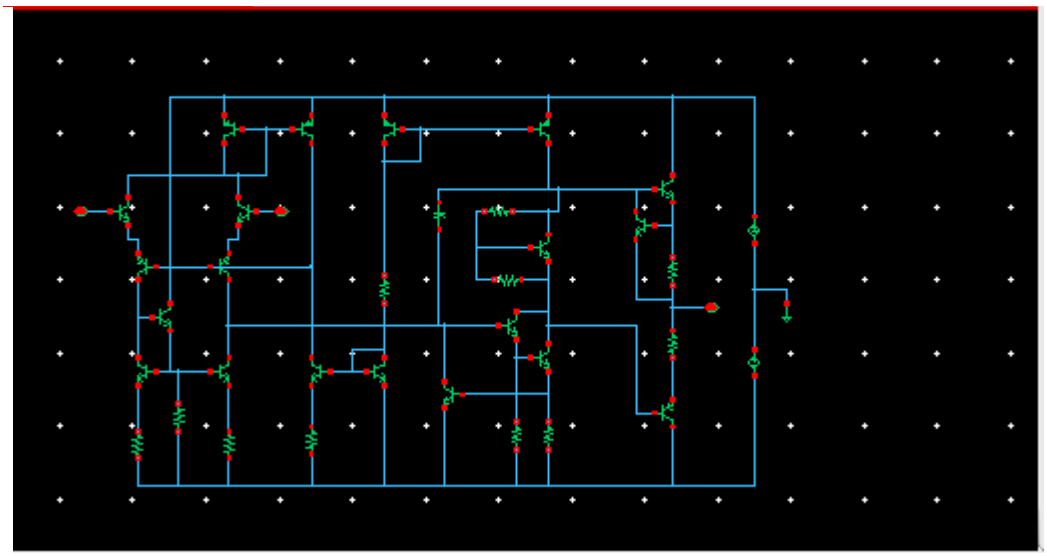


© 2023 Cadence Design Systems, Inc.

You can use the context-menu options to add, delete, or rename columns.

You can also define performance specifications in the Outputs Setup pane in the *Spec* column. After simulation, the results for the specified specifications are displayed in the *Value* column to indicate whether your design met, nearly met, or failed to meet your performance specifications. The *Value* column displays simulation status, such as pending, running, error codes, and so on, and the text foreground color is changed based on error code of the result. A waveform icon is displayed in this column when the output result is a wave. When a run is multi-point (sweep), this column is not displayed.

When you click the tab for the design schematic (*I19 (opamp)*), the Virtuoso Schematic Editor window opens and design appears as shown in the figure below:



You can close these tabs to exit from the schematic and ADE Explorer environment.

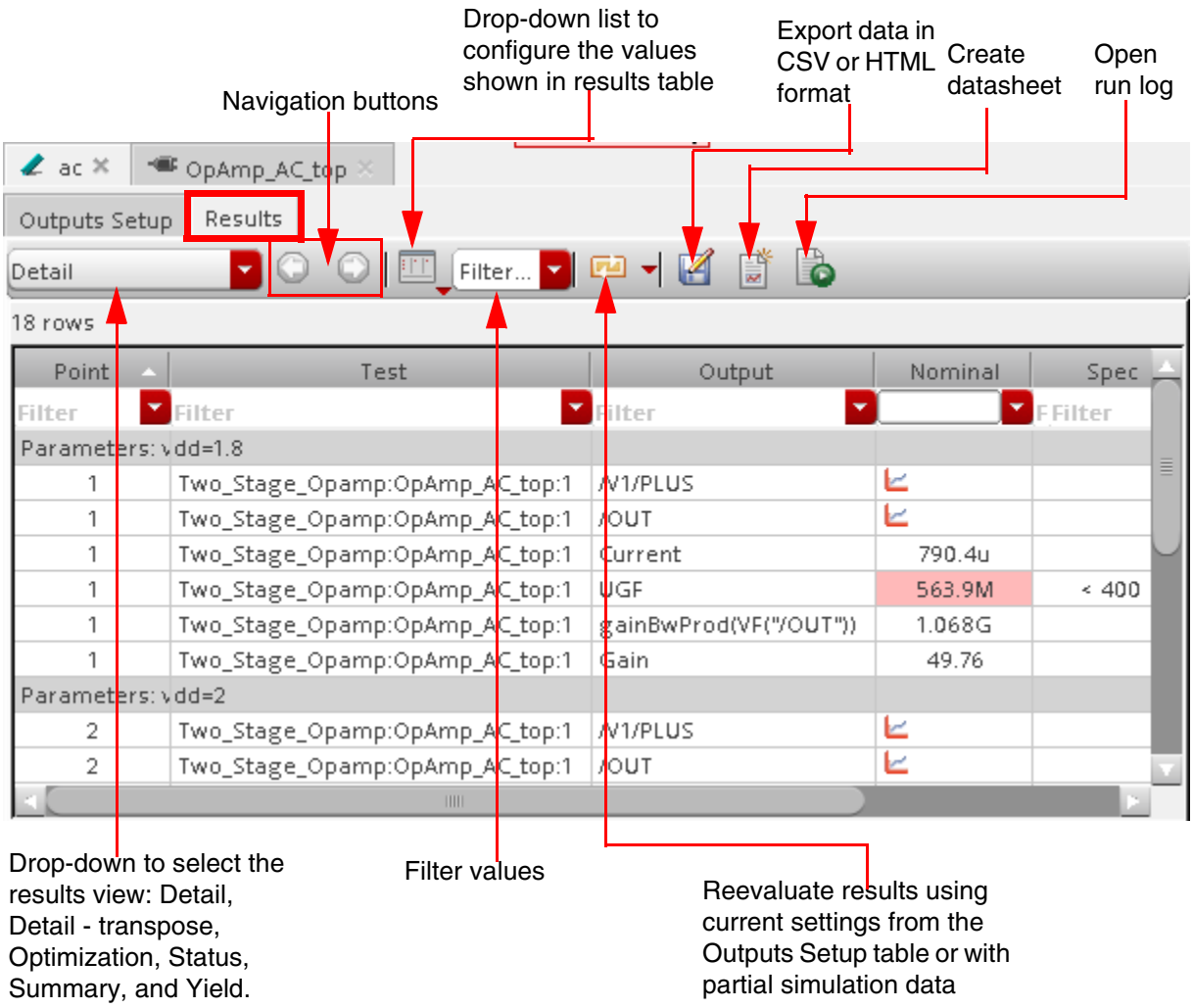
## Results Tab

The Results tab is displayed when you run a simulation with the following features:

- Multi-point simulation with sweep values defined for parameters or corners
- Monte Carlo Sampling
- Reliability Analysis
- Checks/Asserts analyses


© 2023 Cadence Design Systems, Inc.

The Results tab displays the results for each swept value separately (see the figure below).




After running a multi-point simulation, if you run a single-point simulation, the *Result* tab disappears and the results are displayed in the *Value* column of the *Outputs Setup* tab.

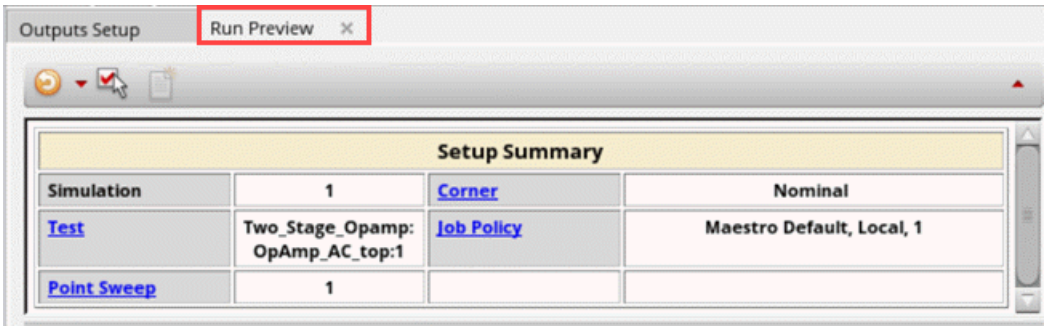
## Run Preview Tab

### Run Preview

The *Run Preview* tab in the Outputs Setup pane provides a summarized view of all the settings applied in the current ADE Explorer view. You can view various details, such as the total number of sweep points and corners that are run, the number of active tests, or the job policies being used. To view specific data points, use the filters available for each column


© 2023 Cadence Design Systems, Inc.

header. If required, you can also view a particular data point in the debug environment and run simulation for that.




For more information about the Run Preview tab, see Viewing Run Preview on page 639.

## Status Bar

The status bar displayed at the bottom of the window displays the following information:

- Simulation progress during the simulation run, as shown in the figure below:

- Results history name, design, and simulator information
- Warnings and error messages
- Static information, such as the name of the toolbar button selected in the window
- Dynamic information, such as the toolbar names are displayed when you perform mouse-hover on toolbars


© 2023 Cadence Design Systems, Inc.

# Switching Between ADE Explorer and ADE Assembler

Video

For an overview of this feature, see Transitioning between ADE Assembler and Explorer video on Cadence Online Support.

In ADE Explorer, you can work on one test at a time. When you need to work on different tests simultaneously, you can launch the Virtuoso ADE Assembler application that provides a multi-test environment where you can set up and run multiple tests.

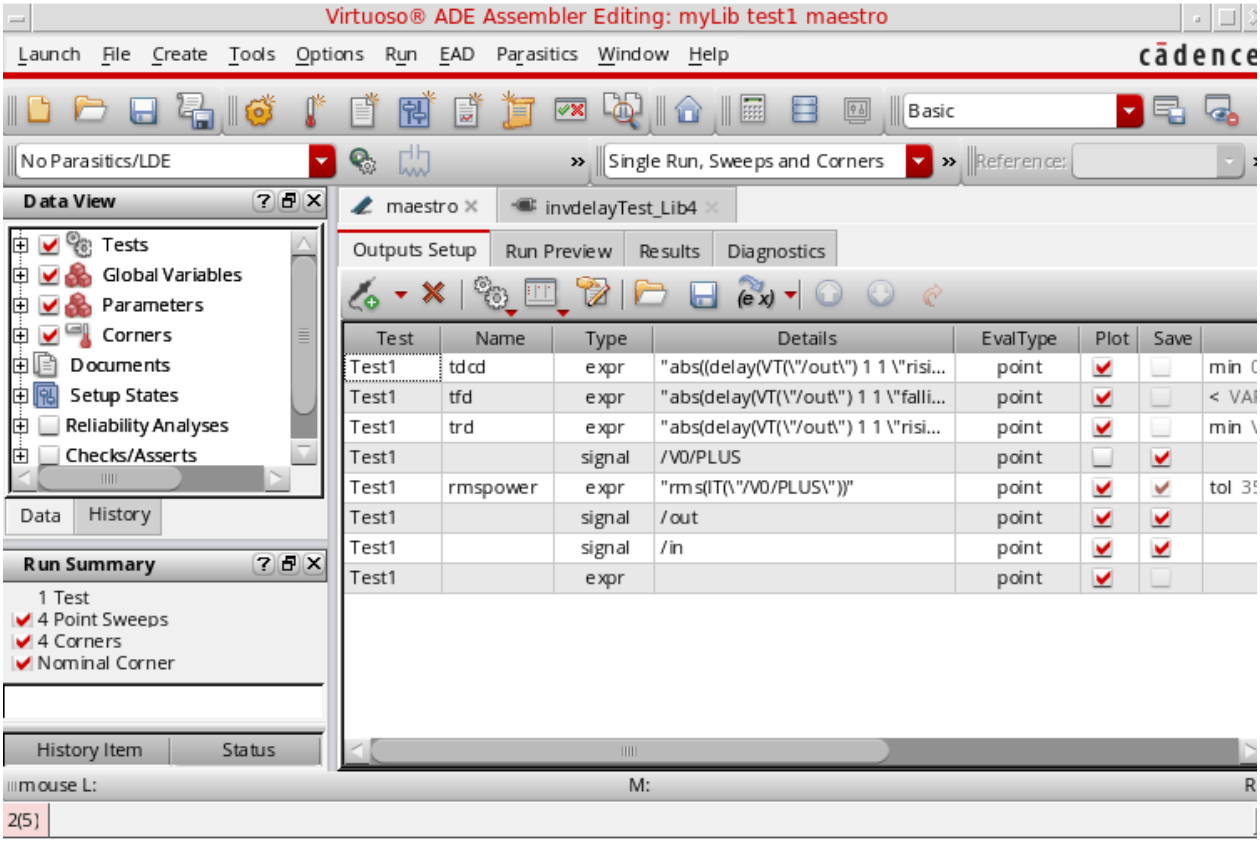
*Important*

To launch Virtuoso ADE Assembler, you must have a valid Assembler license.

To launch Assembler from within ADE Explorer,

➡ Choose *Launch – ADE Assembler*.

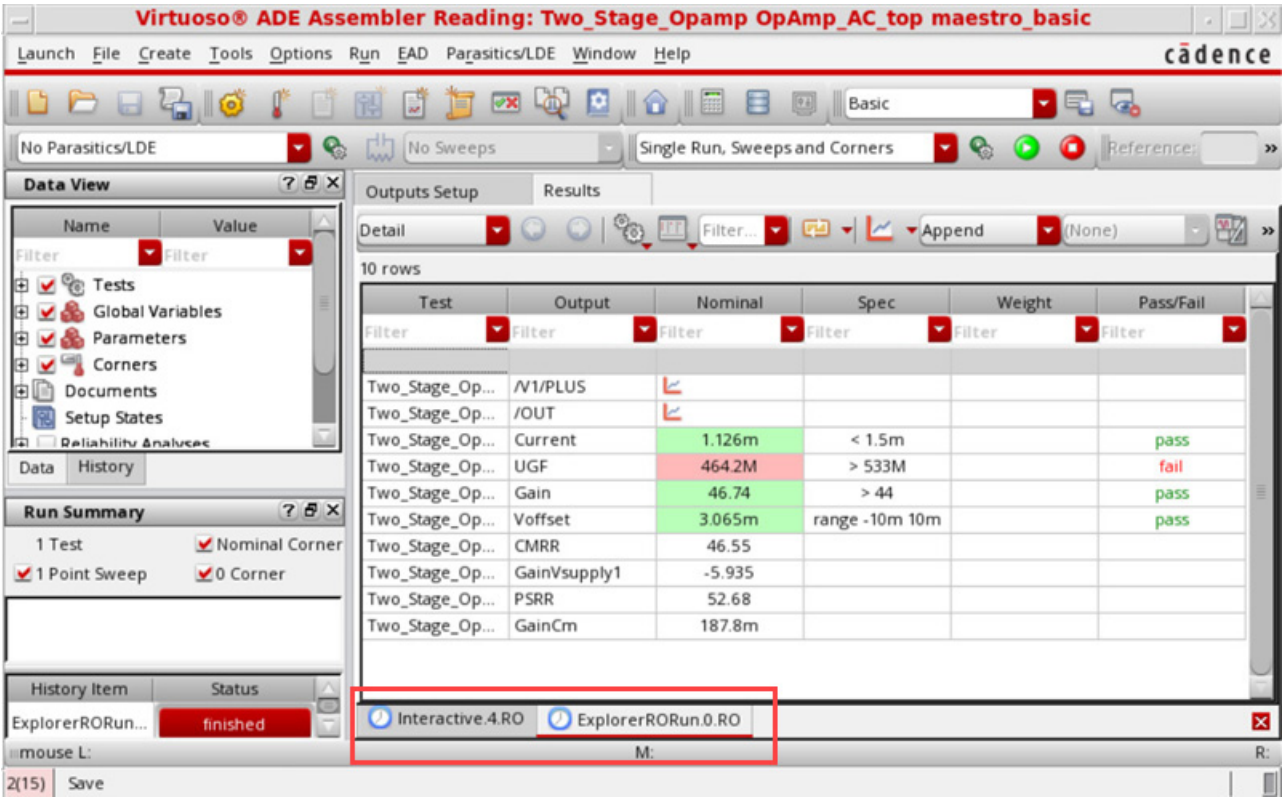
The current ADE Explorer setup appears in the Virtuoso ADE Assembler environment.



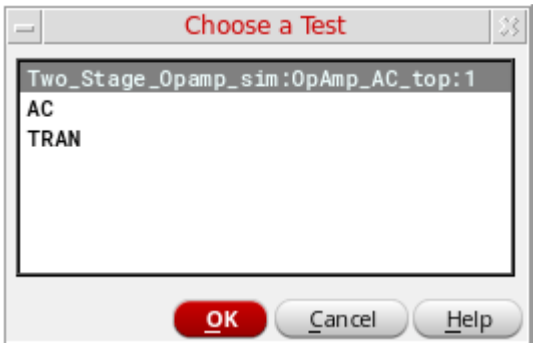


© 2023 Cadence Design Systems, Inc.

**Note:** After switching from ADE Assembler to ADE Explorer, if you run a simulation and then move back to ADE Assembler, the history tabs that were initially created in ADE Assembler are retained. Any histories created in ADE Explorer will also be visible in ADE Assembler.




For information about how to work in the ADE Assembler environment, see *Virtuoso ADE Assembler User Guide*. To go back to ADE Explorer from within ADE Assembler, choose *Launch – ADE Explorer*. If you have multiple tests in ADE Assembler, the Choose a Test form appears in which you can choose the test you want to open in ADE Explorer.




This form also appears when you open a setup created in Virtuoso ADE Assembler into Explorer that has multiple tests defined.


© 2023 Cadence Design Systems, Inc.

**Note:** While switching from ADE Assembler to ADE Explorer, if the run mode specified in Assembler is not supported in ADE Explorer (for example, Worst Case Corners, High Yield Estimation), the run mode in ADE Explorer is set to `Single Run, Sweeps and Corners`.

When EAD is enabled for a test in ADE Assembler and you open this test in ADE Explorer, the following error message appears:

```
EAD is not supported in ADE Explorer. Do you want to disable EAD and proceed with
the simulation?
```


© 2023 Cadence Design Systems, Inc.

© 2023 Cadence Design Systems, Inc.

# 2

# Setting Up Explorer Environment

The Virtuoso ADE Explorer lets you can run different simulations for a given design. You choose a simulator and the types of analyses you want to run for the design. You can also specify design variables that you want to use for simulations. When you run simulation, the results are saved in a predefined formats and can be used by other tools, such as the Virtuoso Visualization and Analysis XL for further analysis. If there are signals in the simulation results, they are by default plotted in the Virtuoso Visualization and Analysis XL graph window.

This chapter covers the following topics that describe how you can perform the basic environment settings in Explorer, which includes setting up simulator options, simulation files, environment options, design variables, and so on. It also explains how to configure and customize the default Explorer settings.

ADE Explorer contains a single test and the test name is displayed at the top of the Setup assistant. This assistant displays the setup information for the test, such as simulator, analysis, design variables, parameters, corners, Monte Carlo. By default, the test name includes the design information in the `lib:cell:view` format. To change the test name:

➥ Click the test name to select it and then click again. A cursor appears in the test name field. Specify a new name and press `Enter`.

This chapter covers the following topics:


- Prerequisites For Running Explorer on page 95
- Specifying Simulator Options on page 102
- Setting up High-Performance Simulation Options on page 140
- Setting Up Simulation Files on page 155
- Setting up Simulator Environment Options on page 164
- Setting up Design Variables on page 169
- Adding Notes on page 180
- Creating and Displaying Netlists on page 181



© 2023 Cadence Design Systems, Inc.

- Variables for Customizing Netlist Generation on page 185
- Setting up Stimuli on page 189
- Setting up Cosimulation Options on page 211
- Setting up MDL Mode on page 213
- Saving the Current Setup on page 215
- Importing a Saved Setup on page 218
- Saving Results in ADE Explorer on page 229
- Saving an OCEAN Script on page 234
- Configuring and Customizing ADE Explorer Settings on page 237
- Using Signals or Triggers on page 242
- Reserved Words on page 243


© 2023 Cadence Design Systems, Inc.

# Prerequisites For Running Explorer

Before you start performing the Explorer setup and run simulations, you need to perform the following mandatory settings:

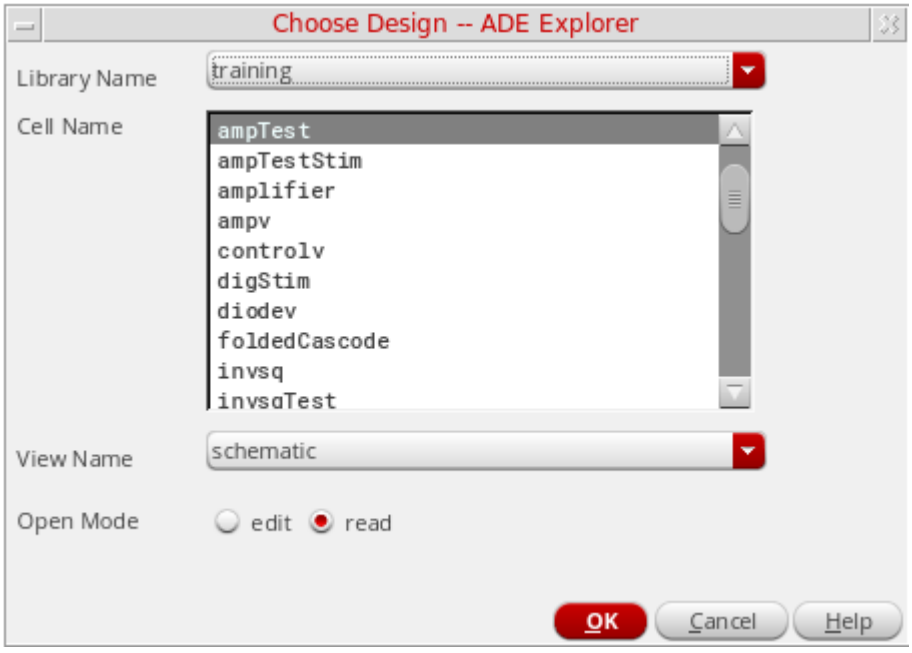
- Choosing a Design on page 95
- Choosing a Simulator on page 96
- Setting the Simulation Temperature on page 97
- Choosing an Analysis on page 98
- Choosing Model Library Path on page 99

## Choosing a Design

To choose a design (library, cell, and view) for a given testcase:

- In the Explorer window, choose *Setup – Design*.
- Right-click the test name and choose *Design*.

  The *Choosing Design -- ADE Explorer* form appears.




In this form,


© 2023 Cadence Design Systems, Inc.

1. In the *Library Name*, *Cell Name*, and *View Name* fields, choose library, cell, and view names.
2. In the *Open Mode* field, choose the mode in which you want to open the selected design —*edit* or *read* mode, and click *OK*.

The Choosing Design form also appears when you create a new maestro view and open Explorer using CIW and Schematic window. For more information about opening Explorer using these methods and choosing a design, see Opening Virtuoso ADE Explorer on page 37.

You can also open an existing design and state in ADE Explorer that was created using ADE L. For more details, see Appendix E, “Migrating From ADE L to ADE Explorer”.

## Choosing a Simulator

The default simulator used to run simulations in Explorer is Spectre. To use another simulator, do the following:

1. Choose *Setup – Simulator*.
2. Double-click in the *Simulator* section of the Setup assistant.
3. Right-click the test name and choose *Simulator*.

   The Choosing Simulator form appears.

4. In the *Simulator* drop-down list, select the simulator you want to set. The available options are—`spectre`, `hspiceD`, `ultraSim`, and `ams`.

*Important*

When you switch from one simulator to another, a new oasis session is created. The new simulator retains the setup files recognized by it and ignores the rest. This behavior is controlled by *retainStateSettings* environment variable with the default value “`yes`”. On setting this variable to “`no`”, the new simulator does not retain the setup files from the previous oasis session.

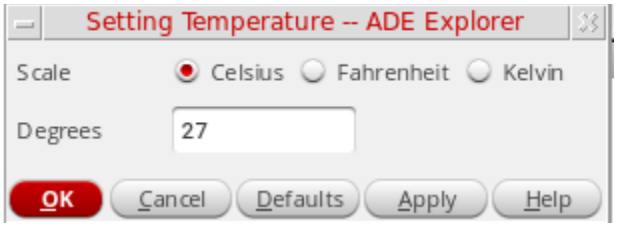

© 2023 Cadence Design Systems, Inc.

## Setting the Simulation Temperature

To set the temperature for the simulation run:

1. Choose *Setup – Temperature*.
2. Right-click the test name and choose *Temperature*.

   The Setting Temperature form appears.




   Alternatively, click the temperature icon ( ) and specify the temperature value in the text box.

3. In the *Scale* field, select the unit you want to use to specify temperature.
4. In the *Degrees* field, specify the temperature value.
5. Click *OK*.

To sweep the temperature,

➡ Specify a design variable named `temperature` in the Setup assistant and add sweep values in the `start:step:end` format.

*Important*

Temperature sweep cannot be specified using the `Temperature` toolbar and *Setting Temperature - ADE Explorer* form. When you add the `temperature` design variable, this toolbar and *Setup – Temperature* menu are disabled.

You can also sweep the temperature by creating a design variable, for example, `var1`, and then in the *Degrees* field, type `VAR("var1")`. The value of variable `var1` can be swept to perform a temperature sweep. While specifying the temperature value through the `VAR` function, if you have a temperature value specified in the Corners Setup form, an error message is displayed asking you to confirm whether the values in the Corners Setup form should be removed or retained.


© 2023 Cadence Design Systems, Inc.

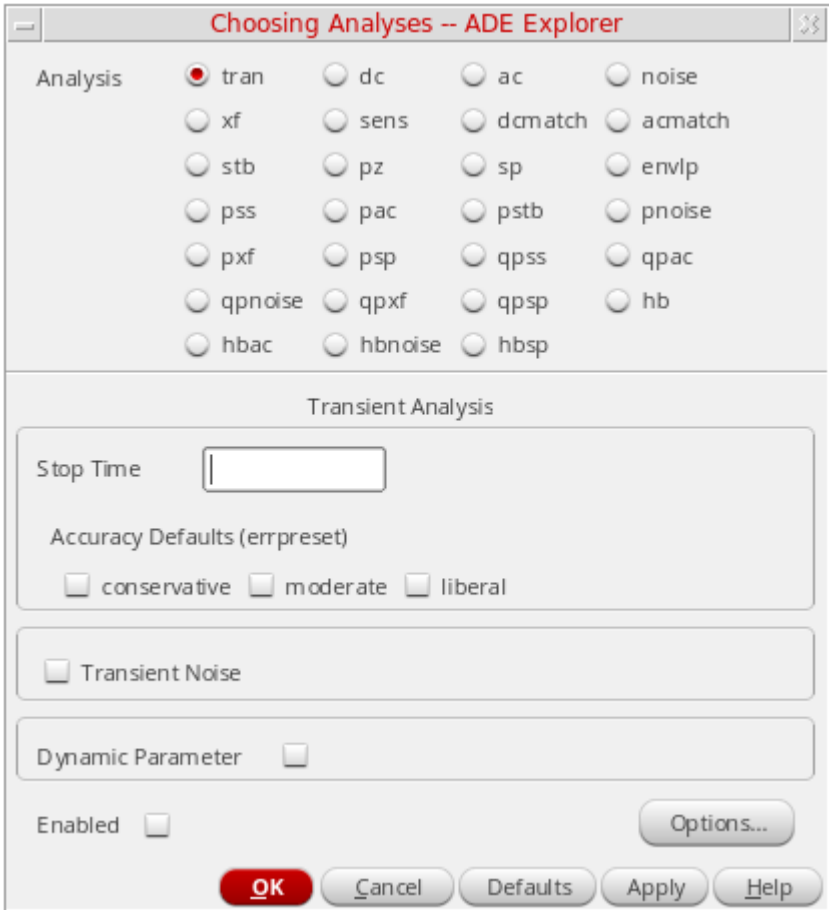
## Choosing an Analysis

After you specify a simulator, you need to choose the analysis type on which you want to run simulation. These analysis type vary according to the simulator you select.

To select an analysis type:

- Choose *Analyses – Choose*.
- Click the `Click to add analysis` in the Setup assistant.
- Click the Choose Analysis icon on the Run toolbar.
- Right-click the test name and choose *Add Analysis*.

  The *Choosing Analyses -- ADE Explorer* form is displayed.




In this form, select the analysis you want to set and specify its attributes. For detailed information about the analyses in Spectre, see Setting Up Analyses.


© 2023 Cadence Design Systems, Inc.

**Note:** If the simulation performance mode is set to Spectre FX, only the *tran*, *dc*, and *ac* analysis options are available in the *Choosing Analyses -- ADE Explorer* form.

## Choosing Model Library Path

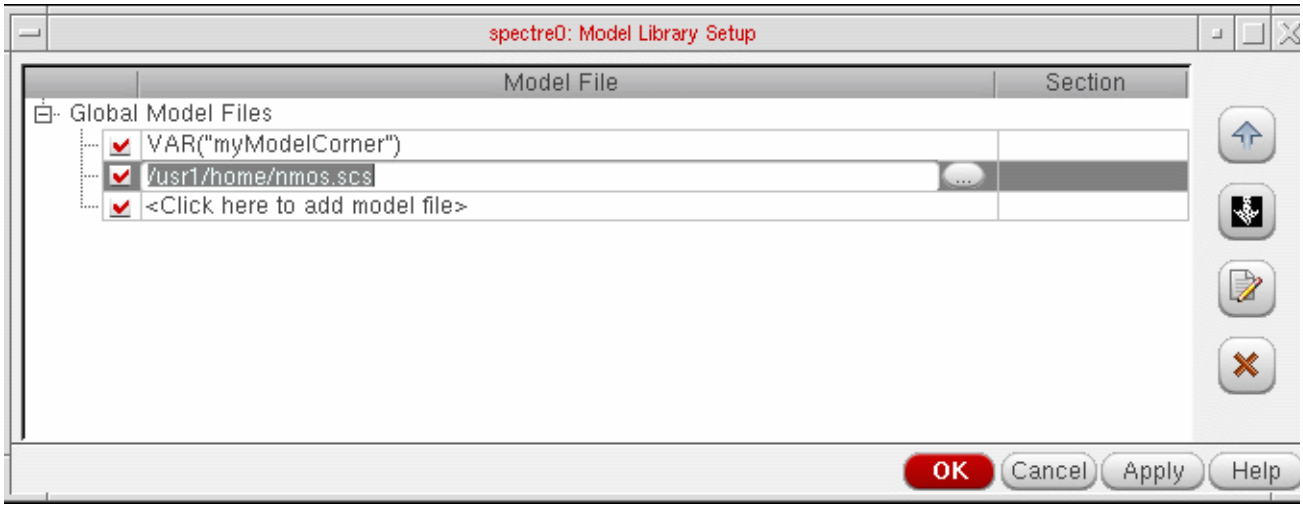
To set up models libraries for simulator interfaces, do one of the following:

- In the ADE Explorer window, choose *Setup – Model Libraries*.
- Right-click the test name and choose *Model Libraries*.

  Note the following when you are entering the path and filename:

  - You can use absolute and relative paths.
  - You can use environment variables to specify the path.
  - You can use design variables to specify the model library name.

  The *Model Library Setup* form appears.




- In the *Global Model Files* column, type the path and filename of the model file you want to use.

  The model files contain all model definitions referred to by your design and not defined within the Virtuoso library. Unless you specify a full path, the simulator assumes that you are using the project directory.

  Alternatively, you can click the browse button to select a valid model file.Alternatively, you can click the browse button to select the model library files to be added.

  **Note:** To add multiple model library files, press the *SHIFT* or *CTRL* key and click the


© 2023 Cadence Design Systems, Inc.

files you want to add.

**Note:** By default, when model files are added, the symbolic links in the file paths are resolved so that path to the file in the design cache is shown in the GUI.

- (Optional) In the *Section* column, select a section from the drop-down list.
- If the check box next to a model library is selected, the model library is enabled for the design. To disable the model library, clear the check box next to the library.

  You can also add, enable or disable model library files using the `asiSetEnvOptionVal` SKILL functions as shown below:

```
asiSetEnvOptionVal(asiGetTool('spectre) "modelFiles"
list(
   list("/usr1/models/model1.scs")
   list("/usr1/models/model2.scs")
   list("#" "/usr1/models/model3.scs")
))
```

  The `#` symbol is used to disable the `model3.scs` file. If the # symbol is not used, the model library is enabled for the design.

**Note:** If you use an empty string, that is `" "`, as the first argument, the specified model file will not be included in the model library.

The order in which the libraries are listed in the form determines their search order. Libraries are searched starting at the top of the list. If a model is included in two or more libraries, you can change the search order to determine which library ADE Explorer searches first. ADE Explorer uses the first model found in the search order.

- To move a library one level up, select the library and click the button.
- To move a library one level down, select the library and click the button.
- To edit a model library file, select the file and click the button to open the file in a text editor
- To delete a model library file, select the file and click the button.

**Example:**

The model file for a direct interface simulation of the schematic view of the lowpass cell of the `aExamples` library can be found in *`your_install_dir`*`/tools/dfII/samples/artist/models/spectre/definitions.scs`

```
simulator lang=spectre
model npn bjt type=npn is=3.26E-16 va=60 bf=100 \
br=6 nc=2 ikr=100m rc=1 vje=0.7 \
```


© 2023 Cadence Design Systems, Inc.

```
cjc=1e-12 fc=0.5 cje=0.7e-12 \
tr=200e-12 tf=25e-12 itf=0.03 vtf=7 xtf=2

model pnp bjt type=pnp is=3.28e-16 va=30 bf=35 \
br=6 nc=2 ikr=100m rc=1 \
cjc=1e-12 fc=0.5 cje=0.7e-12 \
tr=200e-12 tf=65e-12 itf=0.03 vtf=7 xtf=2
```

The models `npn` and `pnp` are referenced within the `opamp` schematic cell-view of the `aExamples` library. The device `Q25` (connected to the pin `inp`) references the model `npn` with the parameter `model` (Model Name).


© 2023 Cadence Design Systems, Inc.

# Specifying Simulator Options

This section describes the various simulator that are supported in Explorer. The default simulator is Spectre.

- Spectre
- Spectre X
- Spectre FX
- Virtuoso Accelerated Parallel Simulator (APS)
- Virtuoso UltraSim Simulator Interface
- Virtuoso AMS Simulator Interface
- Virtuoso HSPICE Simulator Interface

## Spectre

The Explorer environment provides the spectre interfaces to the Spectre® analog simulator.

You need to use the MMSIM version of Spectre, which is available on the MMSIM CD and not in the DFII hierarchy. If you set up your path to point to a previous (non-MMSIM) version of the Spectre software, the simulation will not run and you get the following message:

```
"The 'spectre' executable that you are using is an older version.Use MMSIM60 or
later version of Spectre with this release. To check the spectre version, run
'spectre -W'."
```

The Spectre simulator is integrated into the analog circuit design environment with the Open Analog Simulation Integration Socket (OASIS).

To run real time tuning simulations using Spectre, use MMSIM14.1ISR3 or beyond, otherwise the RTT simulations are run in the batch mode. For more information, see Chapter 10, “Real-Time Tuning in Simulations”.

### Setting Spectre Options

Perform the following steps to set the simulator-specific options and variables for Spectre:


© 2023 Cadence Design Systems, Inc.

- Choose *Simulation – Options – Analog*. This command helps you set simulator options and variables that apply to all analyses.

© 2023 Cadence Design Systems, Inc.

You can turn on the AHDL linter feature when the Fast APS mode is enabled. For information about the AHDL Linter feature, refer to the chapter AHDL Linter Checks in the Spectre Classic Simulator, Spectre APS, Spectre X, and Spectre XPS User Guide.

# Simulator Options Form

The Simulator Options form lets you specify simulator-specific options and variables that apply to all analyses.

The form contains the following tabs.

| Tab | Description |
|---|---|
| Main | Lets you specify the values for toleration, temperature, multithreading, noise, and high voltage options. |
| Algorithm | Lets you specify the values for convergence and matrix parameter options. |
| Component | Lets you specify the values for component, nport, and model options. |
| Check | Lets you specify the values for error checking and device checking options. |
| Annotation | Lets you specify the values for annotate options. |
| Miscellaneous | Lets you specify the values for quantity, sensitivity, AHDL linter, and other miscellaneous options. |

## Main

The following table describes the fields available on the *Timestep* tab of the Simulator Options form.

| Field | Description |
|---|---|
| ***TOLERATION OPTIONS*** | Specifies the values for toleration options. |
| *reltol* | Specifies the relative convergence criterion.<br>The default value is `1e-3`. |
| *residualtol* | Specifies the time after which the outputs are to be saved. |


© 2023 Cadence Design Systems, Inc.

| Field | Description |
|---|---|
| *vabstol* | Specifies the convergence criterion for absolute voltage tolerance.<br>The default value is `1e-6`. |
| *iabstol* | Specifies the convergence criterion for absolute current tolerance.<br>The default value is `1e-12`. |
| ***TEMPERATURE OPTIONS*** | Specifies the values for temperature options. |
| *temp* | Specifies the temperature.<br>The default value is `27`. |
| *tnom* | Specifies the temperature measurement of the default component parameter.<br>The default value is `27`. |
| *tempeffects* | Specifies the temperature effect selector.<br>■ `vt`: Varies only the thermal voltage with temperature.<br>■ `tc`: Specifies that the parameters that start with tc are active and the thermal voltage is dependent on temperature.<br>■ `all`: Enables all built-in temperature models.<br>This is the default. |
| ***MULTITHREADING OPTIONS*** | Specifies the values for multithreading options. |
| *multithread* | Enables or disables the multithreading capability. When multithreading is enabled but the number of threads (*nthreads*) is not specified, Spectre automatically detects the number of processors and selects the proper number of threads to use.<br>■ `on`: Enables multithreading.<br>■ `off`: Disables multithreading. |
| *nthreads* | Specifies the number of threads for multithreading.<br>The default value is `1`. |
| ***NOISE OPTIONS*** | Specifies the values for time step parameters. |


© 2023 Cadence Design Systems, Inc.

| Field | Description |
|---|---|
| *Noise Contribution* | Enables or disables the noise contribution for the given instance list.<br>■ `on`: Enables the noise contribution for the given instance list.<br>■ `off`: Disables the noise contribution for the given instance list. |
| *Instance List* | Specifies the list of instances to be considered for noise contribution.<br>**Note:** This option is available only when *Noise Contribution* is set to `on`. |
| *Select* | Lets you select instances from the schematic.<br>**Note:** This option is available only when *Noise Contribution* is set to `on`. |
| *Clear* | Clears the selected instances from the instance list.<br>**Note:** This option is available only when *Noise Contribution* is set to `on`. |
| *noiseon_type* | Defines the noise sources that are enabled by the `noiseon/off_inst` option.<br>Possible values are `thermal`, `flicker`, `shot`, `ign`, and `all`.<br>The default value is `all`. |
| *noiseoff_type* | Defines the noise sources that are disabled by the n`oiseon/off_inst` option.<br>Possible values are `hermal`, `flicker`, `shot`, `ign`, and `all`.<br>The default value is `all`. |
| ***HIGH VOLTAGE*** | Specifies the values for time step parameters. |
| *highvoltage* | Enables optimized Spectre settings for high voltage designs including voltage, and current binning, excluding VerilogA and dangling nodes from convergence checks, and optimized large capacitance handling.<br>Possible values are `on` and `off`.<br>The default value is `off`. |


© 2023 Cadence Design Systems, Inc.

## Algorithm

The following table describes the fields available on the *Algorithm* tab of the Simulator Options form.

| Field | Description |
|---|---|
| ***CONVERGENCE OPTIONS*** | Specifies the values for initial condition parameters. |
| *homotopy* | Specifies the method used when no convergence occurs on the initial attempt of DC analysis. You can specify methods and their orders by specifying a vector setting such as `homotopy=[source ptran gmin]`.<br>Possible values are `none`, `gmin`, `source`, `dptran`, `ptran`, and `all`.<br>The default value is `all`. |
| *limit* | Specifies the limiting algorithms to aid DC convergence.<br>Possible values are `delta`, `log`, `dev`, `l2`, `newton`, and `none`.<br>The default value is `dev`. |
| *gmethod* | Stamps gdev, gnode, or both in the homotopy methods (other than dptran).<br>Possible values are `dev`, `node`, and `both`.<br>The default value is `dev`. |
| *dptran_gmethod* | Stamps gdev, gnode, or both in the dptran (homotopy) methods.<br>Possible values are `dev`, `node`, and `both`.<br>The default value is `node`. |
| *try_fast_op* | Speeds up the DC solution. For hard-to-converge designs, this feature fails and other methods are applied. In corner cases, this feature may have negative effects. If the DC analysis is unusually slow, the memory usage of the processes keeps increasing, or if DC analysis gets stuck even before homotopy methods start, set this option to `no`.<br>Possible values are `yes` and `no`.<br>The default value is `yes`. |


© 2023 Cadence Design Systems, Inc.

| Field | Description |
|---|---|
| *convdbg* | Specifies theoption to diagnose convergence issues and identify problem areas.<br><br>Possible values are `none`, `status`, and `detailed`. |
| *icpriority* | Sets the IC priority order. If set to netlist, the order from lowest to highest is readNS, netlist NS, readIC, netlist IC. If set to file, the order from lowest to highest is netlist NS, readNS, netlist IC readIC.<br><br>Possible values are `file` and `netlist`.<br><br>The default value is `netlist`. |
| *nonconv_topnum* | Specifies the top number of non-convergence nodes to be printed public.<br><br>The default value is `10`. |
| *gmin* | Specifies the conductance for each nonlinear branch of the device to prevent simulation non-convergence. A Large `gmin` impacts accuracy of the current probe, while a small `gmin` value may cause circuit convergence issue. For a circuit that is sensitive to leakage current, it is recommended to set `gmin` to a small value or zero.<br><br>The default value is `1e-12`. |
| *gmin_check* | Specifies that the effect of `gmin` should be reported if significant.<br><br>Possible values are `no`, `max_v_only`, `max_only`, and `all`.<br><br>The default value is `max_v_only`. |
| *gmindc* | Specifies the minimum conductance across each non-linear device in DC analysis. If `gmindc` is not specified, the `gmin` value is taken as the value for `gmindc`.<br><br>The default value is `1.0e-12`. |
| *rforce* | Specifies the resistance used when forcing nodesets and node-based initial conditions.<br><br>The default value is `1`. |


© 2023 Cadence Design Systems, Inc.

| Field | Description |
|---|---|
| *rthresh* | Specifies that all instance resistors with `abs(R) < rthresh` most use resistance form unless their instance parameter or model parameter overwrites it.<br><br>The default value is `0.001`. |
| *rabsclamp* | Specifies that all fixed value and variable resistors with an absolute value of `R<=rabsclamp` must be clamped to `rabsclamp`.<br><br>The default value is `0.0`. |
| *rabsshort* | Specifies that all fixed value resistors with absolute value of `R<=rabsshort` must be shorted.<br><br>The default value is `0` for Spectre, and `1m` for APS.<br><br>`rabsshort` can additionally be applied to variable resistors using the option `short_cut_var_elem=yes`. |
| ***MATRIX PARAMETER OPTIONS*** | Specifies the values for initial condition parameters for oscillators. |
| *pivotdc* | Enables the use of numeric pivoting on every iteration of DC analysis.<br><br>Possible values are `yes` and `no`.<br><br>The default value is `no`. |
| *dc_pivot_check* | Specifies that during a DC analysis, the numeric pivoting must be performed only when bad pivot is detected.<br><br>Possible values are `yes` and `no`.<br><br>The default value is `no`. |
| *pivrel* | Specifies the relative pivot threshold.<br><br>The default is `1e-3`. |
| *pivabs* | Specifies the absolute pivot threshold.<br><br>The default is `0`. |


© 2023 Cadence Design Systems, Inc.

| Field | Description |
|---|---|
| *preorder* | Controls the amount of matrix pre-ordering that is done and may lead to much fewer matrix fill-ins in some cases. Known cases include designs with large number of small resistors and large number of behavioral instances containing voltage based equations. Try this option when the simulation runs out of memory or if the simulation is unreasonably slow for the size of your design.<br>Possible values are `partial` and `full`.<br>The default value is `partial`. |
| *rebuild_matrix* | Ensures consistent matrix ordering at the beginning of the analyses for consistent results.<br>If set to `yes`, rebuild circuit matrix at the beginning of ac, dc, dcmatch, montecarlo, pz, stb, sweep, tdr, and tran analyses. Notice that rebuild circuit matrix can result in performance overhead.<br>Possible values are `yes` and `no`.<br>The default value is `no`. |

## Component

The following table describes the fields available on the *Component* tab of the Simulator Options form.

| Field | Description |
|---|---|
| ***COMPONENT OPTIONS*** | Specifies the values for component options. |
| *scalem* | Specifies the model scaling factor.<br>The default value is `1.0`. |
| *scale* | Specifies the device instance scaling factor.<br>The default value is `1.0`. |


© 2023 Cadence Design Systems, Inc.

| Field | Description |
|---|---|
| *scalefactor* | Specifies the Scale factor for Device Model Technology Scaling. The options parameter `scalefactor` enables device model providers to scale device technology independent of the design dimension scaling done by circuit designers. The resulting device instance scaling is defined by `'scale * scalefactor'`. If the foundry uses a technology scale factor of 0.9 (`scalefactor=0.9`), and the circuit designer uses a design scale factor of 1e-6 (scale=1e-6), then the compounded scaling of the device instance dimension is 0.9e-6. Unlike options parameter `scale`, `scalefactor` cannot be used as a netlist parameter and cannot be altered or used in sweep statements.<br><br>The default value is `1.0`. |
| *approx* | Specifies that the approximate models must be used. Difference between approximate and exact models is less.<br><br>Possible values are `yes` and `no`.<br><br>The default value is `no`. |
| *macromodels* | Determines whether the circuit contains macromodels. At times, setting this parameter to `yes` helps improve performance<br><br>Possible values are `yes` and `no`.<br><br>The default value is `no`. |
| *maxrsd* | Lets you use approximation for drain or source parasitic resistors which are less then `maxrsd`. Applies to bsim3v3 and bsim4 mosfet models. |
| *auto_minductor* | Lets you insert missing mutual inductor coupling automatically.<br><br>Possible values are `yes` and `no`.<br><br>The default value is `no`. |


© 2023 Cadence Design Systems, Inc.

| Field | Description |
|---|---|
| *minr* | Specifies that all parasitic resistors inside devices less than global `minr` must be removed.<br>The order of checking devices is as follows.<br>**1.** Check if resistors are smaller than local `minr`. If yes, check if it is a MOSFET or BJT. If it is a MOSFET, drop the resistor, if it is BJT, clamp to the `minr` value, and give a warning message for both cases.<br>**2.** Check global `minr`, All Parasitic resistors less than global `minr` are removed and a warning message is issued.<br>**3.** If the resistor is not removed and is smaller than 0.001, issue a warning.<br>The default value is `0.0`. |
| ***NPORT OPTIONS*** | Specifies the values for nport options. |
| *nportirreuse* | Lets you reuse impulse responses data for all nport instances.<br>Possible values are `yes` and `no`.<br>The default value is `yes`. |
| *nportirfiledir* | Specifies the directory to which the nport impulse response file is to be written. If it is not specified, the file is written to /`home/<username>/.cadence/mmsim/`. If a relative path is specified, the path is relative to the current working directory. |
| *nportcompress* | Improves the efficiency of S-parameter simulation of large nport files when a certain percentage of the ports is unused, i.e., open or short circuited. Nport compression does not impact simulation accuracy.<br>■ `yes`: Attempts to force compression.<br>■ `no`: Disables compression.<br>If left unspecified, compression is enabled when N>=10 and the ratio of used ports is less than or equal to 0.8.<br>Possible values are `yes` and `no`.<br>The default value is `yes`. |


© 2023 Cadence Design Systems, Inc.

| Field | Description |
|---|---|
| *nportcompressfiledir* | Specifies the directory where the compressed nport S-parameter file is to be written. If unspecified, it is stored in the output directory. |
| *nportunusedportgmin* | Leaves the port open-circuited. A small value loads open-circuited ports with a finite but large resistance. This introduces a small error in the response, but it induces losses which help obtain a passive response.<br><br>The default value is `0`. |
| *nportunusedportrmin* | Leaves the port short-circuited. A small value will insert a small resistance in place of short-circuited ports. This introduces a small error in the response, but it induces losses which help obtain a passive response.<br><br>The default value is `0`. |
| *nportbbsfittedfiledir* | Specifies the directory to which the BBSpice fitted S-Parameter file is to be saved. If not specified, the file is saved to a directory named `BBSpiceOutput`. If a relative path is specified, the path is considered relative to the current working directory. |
| *nport_default_passivity* | Checks and enforces passivity of S-parameter for all nport instances. When set to `disable`, this global option has no effect. If set to a value other than `disable`, all nport elements in the netlist without a value for passivity explicitly set have their passivity argument set to the same value as specified in this global option. If an nport instance already has the passivity option specified, the instance option takes priority, if both are present.<br><br>Possible values are no, `check`, `enforce`, `fit_weak_enforce`, `fit_enforce`, and `disable`.<br><br>The default value is `disable`. |


© 2023 Cadence Design Systems, Inc.

| Field | Description |
|---|---|
| *nport_default_interp* | Nport automatically chooses `bbspice` for analysis, such as pss and the tstab interval of hb and pss, and `linear` for analyses, such as ac, dc, tran, and sp, If nport_default_interp is set to `auto_switch`.<br><br>All nport elements in the netlist that do not have `interp` set will have `interp` set to the value specified in the global option `nport_default_interp`. If an nport instance has the `interp` option explicitly specified, the instance option will take priority over the global option.<br><br>Possible values are `spline`, `rational`, `linear`, `bbspice`, and `auto_switch`.<br><br>The default value is `disable`. |
| *nportbbsversion* | Controls the BBspice version used for S-Parameter fitting.<br><br>■ `1`: The BBspice version `1` is selected<br><br>■ `2`: The BBspice version `2` is selected.<br><br>The default value is:<br><br>■ `2`: If the Spectre 20.1 version is used.<br><br>■ `1`: If the Spectre 19.1 version is used. |
| ***MODEL OPTIONS*** | Specifies the values for model options. |
| *vthmod* | Specifies the Vth output selector.<br><br>■ `std`: Outputs the model equation Vth.<br><br>This is the default.<br><br>■ `vthcc`: Outputs constant current Vth and may impact simulation performance. |
| *ivthn* | Specifies the NMOS Vth current parameter.<br><br>The default value is `0.0`. |
| *ivthp* | Specifies the PMOS Vth current parameter.<br><br>The default value is `0.0`. |
| *ivthw* | Specifies the Width offset for constant current Vth.<br><br>The default value is `0.0`. |


© 2023 Cadence Design Systems, Inc.

| Field | Description |
|---|---|
| *ivthl* | Specifies the Length offset for constant current Vth.<br>The default value is `0.0`. |
| *ivth_vdsmin* | Specifies the minimum Vds in constant current Vth calculation public.<br>The default value is `0.05`. |
| *tmevthmod* | Specifies the TSMC constant vth calculation. By default it is not activated.<br>Possible values are `0` and `1`. |
| *vdsatmod* | Specifies the Vdsat output selector.<br>Possible values are `std`, `gds`, and `slow_gds`.<br>The default value is `std`. |

## Check

The following table describes the fields available on the *Check* tab of the Simulator Options form.

| Field | Description |
|---|---|
| ***ERROR CHECKING OPTIONS*** | Specifies the values for error checking options. |


© 2023 Cadence Design Systems, Inc.

| Field | Description |
|---|---|
| *topcheck* | Checks circuit topology for errors.<br>■ `no`: The topology check is not performed.<br>■ `min`: The topology check is performed and `gmin` is inserted between isolated nodes and ground. A heuristic topology check is not performed.<br>■ `full`: The topology check is performed and `gmin` is inserted between isolated nodes and ground. A heuristic topology check is also performed to find nodes that may be isolated due to the numerical nature of the circuit. For example, nodes isolated by reverse biased diodes in MOSFETS.<br>This is the default.<br>■ `fixall`: Attaches `gmin` to all types of isolated nodes, including the ones detected by the heuristic topology check.<br>■ `errmin`: Similar to `topcheck=min` (`topcheck=full`) but the simulation will stop if floating nodes are found.<br>■ `errful`: Same as `errmin`. |
| *iccheck* | Checks if nodes with initial conditions have capacitive path to ground or connected to ground by vsource. IC for such node is treated as nodeset.<br>Possible values are `no`, `vsource`, `cap`, and `all`.<br>The default value is `all`. |
| *ignshorts* | Silently ignores shorted components.<br>Possible values are `yes` and `no`.<br>The default value is `no`. |
| *diagnose* | Prints additional information that might help diagnose accuracy and convergence problems.<br>Possible values are `no`, `yes`, and `detailed`.<br>The default value is `no`. |


© 2023 Cadence Design Systems, Inc.

| Field | Description |
|---|---|
| *redefinedparams* | Specifies whether parameters can be redefined in the netlist. When set to `warning` or `ignore`, the simulator allows you to redefine parameters in the netlist. However, it honors only the last definition of the redefined parameter. Depending on the value set, the simulator displays warning messages for the redefined parameters or does not display any message. When set to `error`, the simulator does not allow you to redefine parameters in the netlist and displays an error message.<br>Possible values are `error`, `ignore`, `warning`, and `warn`.<br>The default value is `error`. |
| *dspf_subckt_wo_instance_call* | Specifies whether the simulator should error out, give a warning, or ignore when a subcircuit in a DSPF is not instantiated.<br>Possible values are `error`, `ignore`, and `warning`.<br>The default value is `ignore`. |
| ***DEVICE CHECKING OPTIONS*** | Specifies the values for device checking options. |
| *dochecklimit* | Enables or disables assert checks for simulation.<br>Possible values are `yes` and `no`.<br>The default value is `yes`. |
| *checklimitfile* | Specifies the file to which assert violations are written. |
| *checklimitdest* | Specifies the destinations where violations are written.<br>Possible values are `file`, `psf`, `sqldb`, and `both`.<br>The default value is `sqldb`. |
| *checklimitskipsubs* | Specifies the array of subcircuit masters or subcircuit master patterns to be skipped in device checking. Patterns can have any wildcard symbols. |
| *checklimitskipinsts* | Specifies the array of instances to be skipped in device checking. Instances can have any wildcard symbols. |
| *checklimitskipfile* | Specifies the file that contains the subcircuit masters or subcircuit master patterns to be skipped in device checking. Patterns can have any wildcard symbols. |


© 2023 Cadence Design Systems, Inc.

| Field | Description |
|---|---|
| *opptcheck* | Checks operating point parameters against soft limits.<br>Possible values are `yes` and `no`. |
| *Options* | Opens the Device Checking Options form. |

## Annotation

The following table describes the fields available on the *Annotation* tab of the Simulator Options form.

| Field | Description |
|---|---|
| ***ANNOTATE OPTIONS*** | Specifies the values for annotate options. |
| *audit* | Prints thetime required by various parts of the simulator.<br>Possible values are `no`, `brief`, `detailed`, and `full`.<br>The default value is `detailed`. |
| *inventory* | Prints the summary of components used.<br>Possible values are `no`, `brief`, `detailed`, and `full`.<br>The default value is `detailed`. |
| *narrate* | Narrates the simulation.<br>Possible values are `no`, `yes`, and `compact`.<br>The default value is `yes`. |
| *debug* | Prints debugging messages.<br>Possible values are `yes` and `no`.<br>The default value is `no`. |
| *info* | Prints informational messages.<br>Possible values are `yes` and `no`.<br>The default value is `yes`. |


© 2023 Cadence Design Systems, Inc.

| Field | Description |
|---|---|
| *note* | Prints notice messages.<br>Possible values are `yes` and `no`.<br>The default value is `yes`. |
| *maxnotes* | Specifies the maximum number of times a notice is issued per analysis.<br>**Note:** This option has no effect on notices issued as part of parsing the netlist. Use the `-maxnotes` command-line option to control the number of all notices issued.<br>The default value is `5`. |
| *warn* | Displays warning messages.<br>Possible values are `yes` and `no`.<br>The default value is `yes`. |
| *maxwarns* | Specifies the maximum number of times a warning message is issued per analysis.<br>**Note:** This option has no effect on warnings issued as part of parsing the netlist. Use the `-maxwarns` command-line option to control the number of all warnings issued.<br>The default value is `5`. |
| *maxwarnstologfile* | Specifies the maximum number of times a warning message is printed to the log file per analysis.<br>**Note:** This option has no effect on warnings printed as part of parsing the netlist. Use the `-maxwarnstolog` command-line option to control the number of all warnings printed to the log file.<br>The default value is `5`. |
| *maxnotestologfile* | Specifies the maximum number of times a notice message is printed to the log file per analysis.<br>**Note:** This option has no effect on notices printed as part of parsing the netlist. Use the `-maxnotestolog` command-line option to control the number of all notices printed to the log file.<br>The default value is `5`. |


© 2023 Cadence Design Systems, Inc.

| Field | Description |
|---|---|
| *error* | Generates error messages.<br>Possible values are `yes` and `no`.<br>The default value is `yes`. |
| *printstep* | Enables Spectre to print results by equal step defined in .tran statement.<br>Possible values are `yes` and `no`.<br>The default value is `no`. |
| *digits* | Specifies the number of digits used when printing numbers.<br>The default value is `5`. |
| *notation* | Specifies the notation to be used to display real numbers to the screen.<br>Possible values are `eng`, `sci`, and `float`.<br>The default value is `eng`. |
| *cols* | Specifies the width of screen in characters.<br>The default value is `80`. |
| *colslog* | Specifies the width of log file in characters.<br>The default value is `80`. |
| *title* | Specifies the circuit title. |
| *print statistics report* | Prints simulation phase statistic reports.<br>Possible values are `basic` and `detailed`.<br>The default value is `basic`. |


© 2023 Cadence Design Systems, Inc.

## Miscellaneous

The following table describes the fields available on the *Miscellaneous* tab of the Simulator Options form.

| Field | Description |
|---|---|
| ***QUANTITY OPTIONS*** | Specifies the values for quantity options. |
| *value* | Specifies the default value quantity. |
| *flow* | Specifies the default flow quantity.<br>The default value is `l`. |
| *quantities* | Prints quantities.<br>■ `no`: No quantities are printed.<br>This is the default.<br>■ `min`: The simulator prints out all defined quantities.<br>■ `full`: the simulator also prints a list of nodes and their quantities. |
| ***SENSITIVITY OPTIONS*** | Specifies the values for sensitivity options. |
| *sensfile* | Specifies the output sensitivity data file name. |
| *sensformat* | Specifies the format of sensitivity data.<br>Possible values are `tabular` and `list`.<br>The default value is `tabular`. |
| *senstype* | Specifies the type of sensitivity being calculated.<br>Possible values are `partial` and `normalized`.<br>The default value is `partial`. |
| *sensfileonly* | Enables or disables raw output of sensitivity results.<br>Possible values are `yes` and `no`.<br>The default value is `no`. |


© 2023 Cadence Design Systems, Inc.

| Field | Description |
|---|---|
| *sensbinparam* | Specifies the sensitivity for binning models.<br>Possible values are `no`, `uncorrelated`, and `correlated`.<br>The default value is `no`. |
| ***AHDL LINTER OPTIONS (APS Only)*** | Specifies the values for AHDL linter options for Spectre APS. |
| *Linter check* | Enables the AHDL lint checker feature for Spectre APS.<br>Possible value is `on`. |
| *Warning type* | Specifies the type of warning messages to be reported by the simulator.<br>Possible values are `warn`, `error`, and `force`. |
| *Max Warning* | Specifies the maximum number of Verilog-A linter warning messages to be reported by the simulator for each message ID.<br>The default value is `5`. |
| ***OTHER OPTIONS*** | Specifies the values for any other options supported by Spectre simulator. |
| *Additional arguments* | Lets you specify any additional arguments supported by Spectre simulator. |

### *Related Topics*

Environment Variables for Spectre Simulator Options Form.

### *Specifying the PSF Version in Simulator Options*

To specify the PSF version you want to use for writing the simulation results, do the following

➡ In the Simulator Options form, on the  tab, in the *Additional Arguments* field, specify the PSF version in the following format:

```
psfversion="1.0.0"
```

The command above sets the PSF version 1.0.


© 2023 Cadence Design Systems, Inc.

### Specifying Performance and Post-Layout Optimization Options for the Spectre Simulator

With the Spectre simulator, you can run simulation using the baseline Spectre version. You can also enable post-layout optimization to perform parasitic reduction with more tunable and higher compression levels. This option also enables other algorithms that optimize the performance of the DC solution phase and matrix solver specifically for large postlayout circuits, enabling transient and DC analyses with these circuits to be performed faster.

For more information about the baseline Spectre version and post-layout optimization, see the *Spectre Circuit Simulator Reference*

**Note:** Spectre mode is supported in MMSIM 7.0 and later releases and post-layout optimization is supported in IC6.1.7 ISR1 and later releases. APS is supported in MMSIM 7.1 and later releases.

***When a Spectre simulation is running in the interactive mode, do not restore a simulation setup in which the Spectre mode or post-layout optimization is enabled. Restoring such a simulation setup terminates the simulation run.***

To specify performance and post-layout optimization options for the Spectre simulator, do the following:

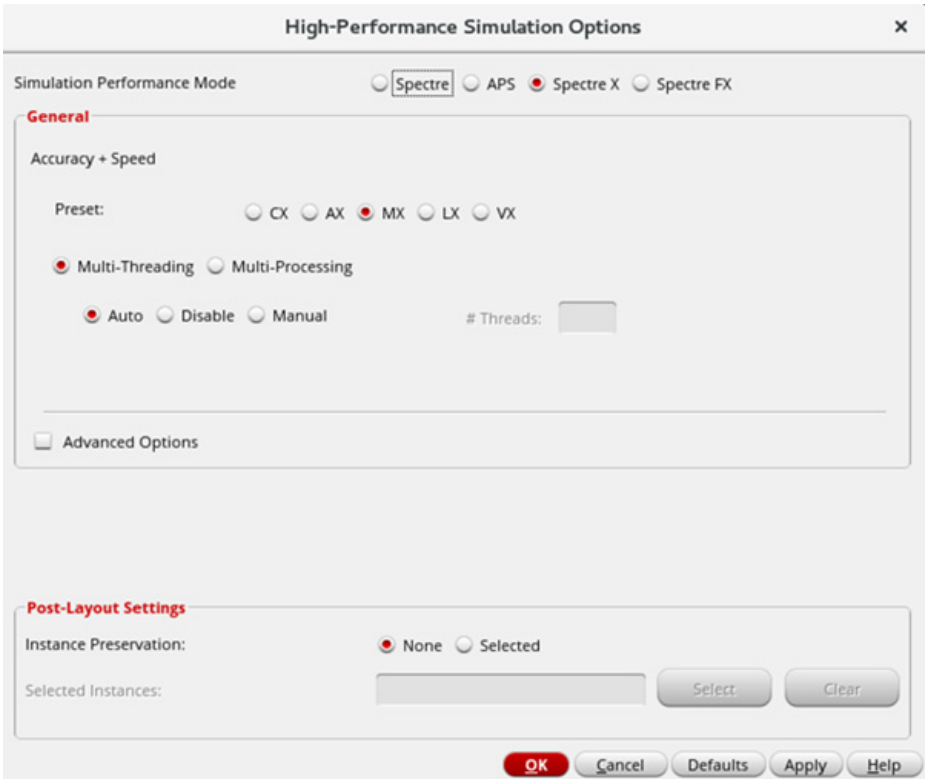
1. Choose *Setup – High-Performance Simulation.*

   The High-Performance Simulation Options form appears.

   **Note:** The *High-Performance Simulation* menu option is disabled when a Spectre


© 2023 Cadence Design Systems, Inc.

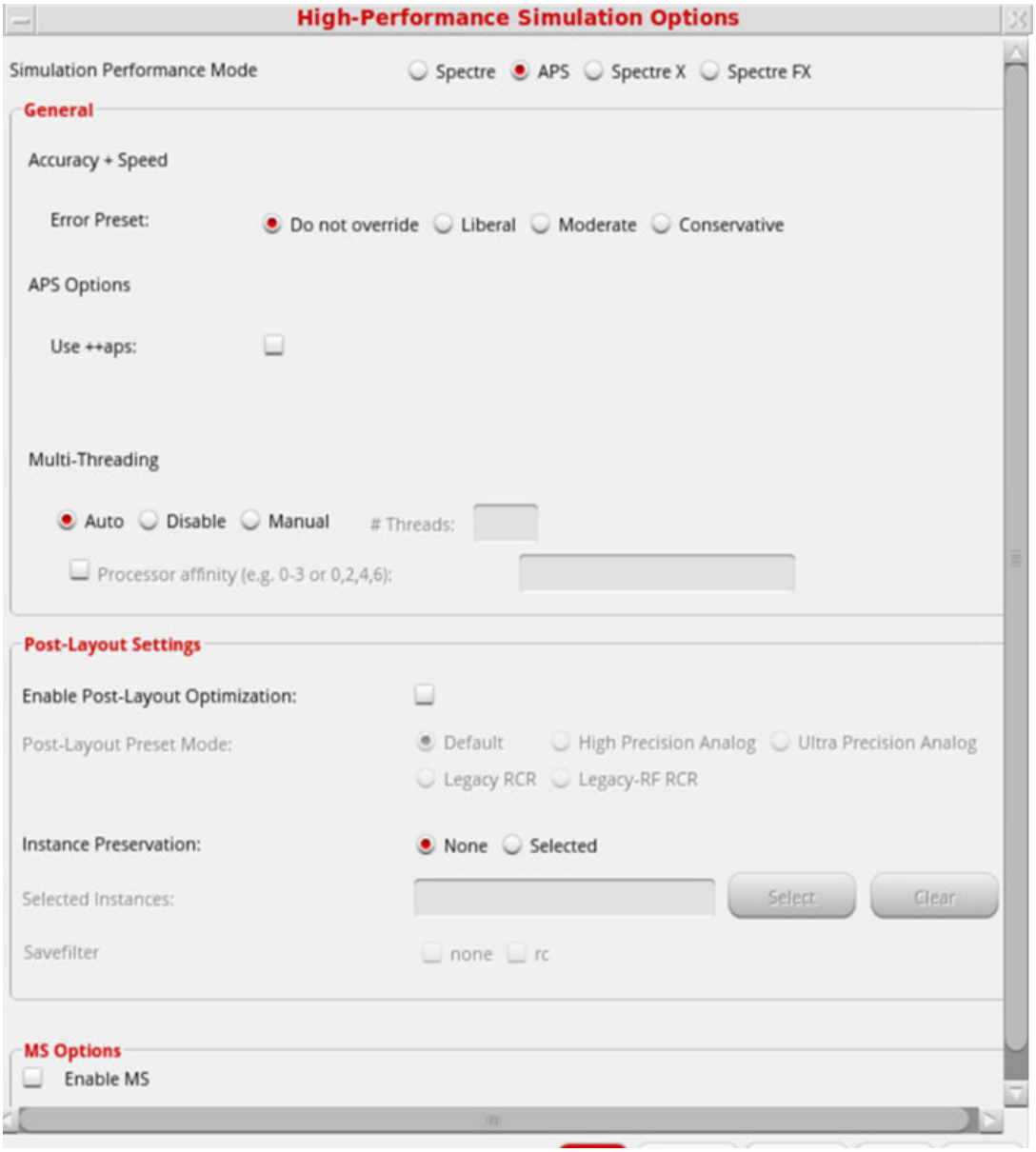
simulation is running in the interactive mode.




2. Select one of the following simulation performance modes:
   - *Spectre*—Use the baseline Spectre version for simulation.


© 2023 Cadence Design Systems, Inc.

- *APS*—Use the Accelerated Parallel Simulator (APS) for simulation. The additional fields required for APS mode appear on the form:




- *Spectre X*— Use the Spectre X simulator mode for simulation. For more information, see Spectre X.

  **Note:** Spectre X mode is supported in Spectre 19.10 release.

- *XPS MS*—Use the Extensive Partitioning Simulator Mixed-Signal (XPS MS) mode for simulation. For more information, see MS Options.

You can switch between Spectre and APS default tolerance options using the *Reset cdsenv options* check box. When the *Simulation Performance Mode* is set to Spectre and


© 2023 Cadence Design Systems, Inc.

the *Reset cdsenv options* check box is selected, ADE resets the transient options to Spectre default settings. If you want to switch to APS default settings, select APS from the *Simulation Performance Mode* and select the *Reset cdsenv options* check box.

**Note:** The default transient options will change from Spectre to APS only if you change the *Simulation Performance Mode* from spectre to APS and select the *Reset cdsenv options* check box and vice versa. If you change the *Simulation Performance Mode* and do not select the check box, the default options will not change.

**Note:** *Reset cdsenv options* check box is available only if the `.cdsenv` file exists in `<cds_inst_dir>/tools/dfII/etc/tools/aps/` location.

3. Specify the accuracy level for transient analysis runs by selecting one of the following options in the *Error Preset* field:

   - Select *Do not override* to use the accuracy level specified for transient analyses runs in the Choosing Analysis form. For more information about setting up a transient analysis, see Transient Analysis on page 250.
   - Select *Liberal*, *Moderate*, or *Conservative* to override the accuracy level specified for transient analyses runs in the Choosing Analyses form.

   If *errpreset* is not set for transient analyses in the Choosing Analyses form and is set to *Liberal*, *Moderate*, or *Conservative* in the High-Performance Simulation Options form, the settings are netlisted and written into the `runSimulation` file.

   For more information about the *Liberal*, *Moderate* and *Conservative* accuracy levels, see the documentation for the `errpreset` parameter in the *Spectre Circuit Simulator Reference*.

   **Note:** crformance mode is set to APS.

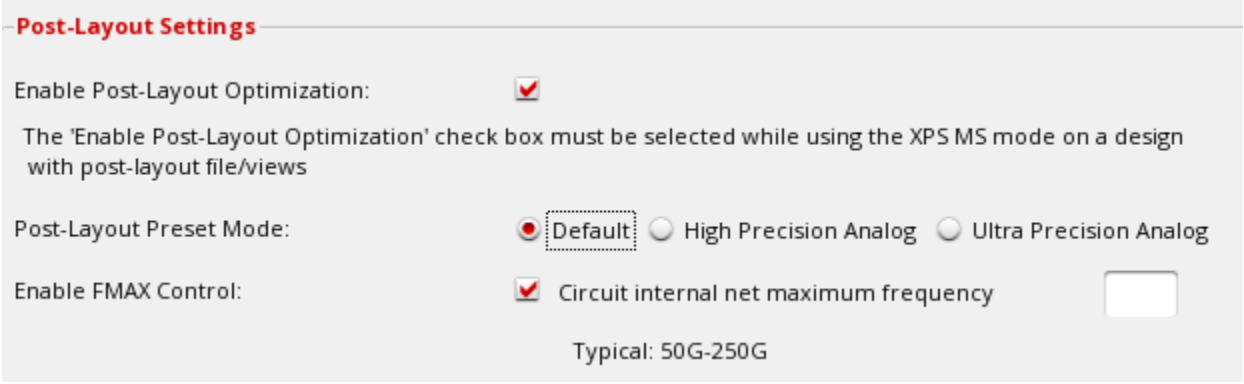
4. Select the *Enable FMAX Control* check box to specify the maximum frequency of the internal net of the circuit or the max operating frequency of the circuit.

   **Note:** This field appears only when the XPS MS simulation performance mode is selected and the *Enable Post-Layout Optimization* check box is selected.

   - *Circuit internal net maximum frequency*—Specify an integer value in this field. It adds the `+rcnet_fmax=<value>` option to the `+postlayout` command line option. The maximum frequency typically ranges from 50G to 250G.
     If the simulation performance mode is set to APS, this field appears when the post-layout preset mode is set to *Default*, *High Precision Analog*, or *Ultra Precision*

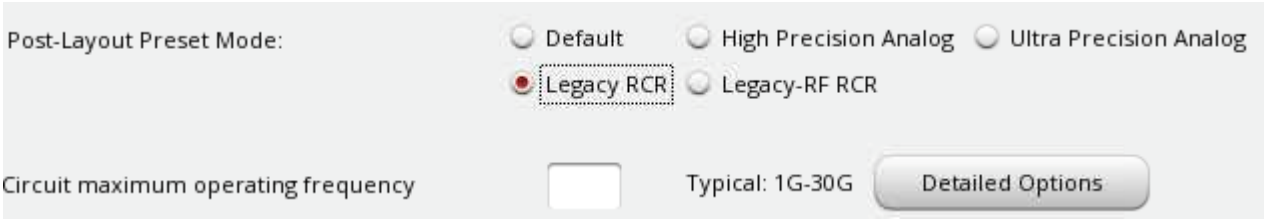

© 2023 Cadence Design Systems, Inc.

*Analog*. And if the performance mode is set to XPS MS, this field appears when the *Enable FMAX Control* check box is selected.




- *Circuit maximum operating frequency*—Specify an integer value in this field. It adds the `rcr_fmax=<value>` global option to the netlist. The operating frequency typically ranges from 1G to 30G.

  This field appears only when the performance mode is set to APS and the post-layout preset mode is set to *Legacy RCR*.



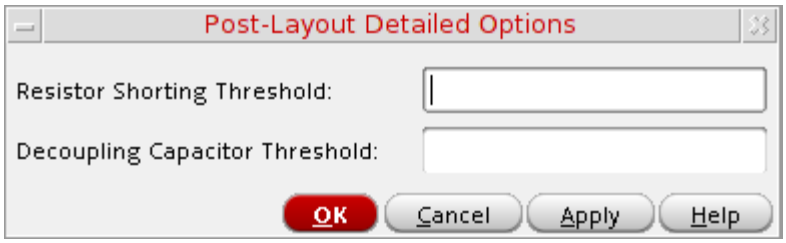

5. Specify the instances to be preserved from being shorted by selecting the required *Instance Preservation* option. Select this check box only if you want to save all operating point data because it might slow down the simulation.

| Select | To |
|---|---|
| *None* | Specify that no instances need to be preserved from being shorted. |


© 2023 Cadence Design Systems, Inc.

| Select | To |
|---|---|
| *Selected* | Specify the instances that must be preserved from being shorted.<br><br>In the *Preserve Instance* field, type the instance names separated by spaces. You can also use wildcards to specify the instances. For example, type `I0.I1.r*` to specify that all instances with names starting with `r` in the `I0.I1` hierarchy must be preserved from being shorted.<br><br>To select instances from the schematic, do the following:<br><br>**1.** Click the *Select* button next to the *Preserve Instance* field.<br><br>The schematic for the design is displayed.<br><br>**2.** Select one or more instances on the schematic.<br><br>To select more than one instance at a time, do one of the following:<br><br>■ Hold down the *Shift* key and click the instances you want to select.<br><br>■ Drag the mouse pointer over the instances you want to select.<br><br>All the instances that are within the yellow bounding box that appears are included in the selection.<br><br>**3.** Press the *Esc* key when you are done.<br><br>The selected instances are displayed in the *Preserve Instance* field. |
| All | Specify that all instances should be preserved from being shorted. |

**6.** (Optional) Click the *Detailed Options* button. The Post-Layout Detailed Options form appears:




In the form, specify the following:

❑ *Resistor Shorting Threshold*— Specify a threshold value (in `ohms`) to short resistors whose resistance is less than the specified value.


© 2023 Cadence Design Systems, Inc.

- *Decoupling Capacitor Threshold*—Specify a threshold value (in `farads`) to decouple capacitors whose capacitance is less than the specified value.

The *Detailed Options* button is enabled only when APS is selected as the performance mode and the *Enable Post-Layout Optimization* check box is selected.

**7.** Click *OK*.

**Note:** If high-performance mode is enabled, the simulator mode is automatically changed to `batch`.

The ADE Explorer window displays the settings you made in the High-Performance Simulation Options form. For example, if you selected the simulation performance mode as and the accuracy level as *Liberal*, the status bar in the window displays the simulator name as:

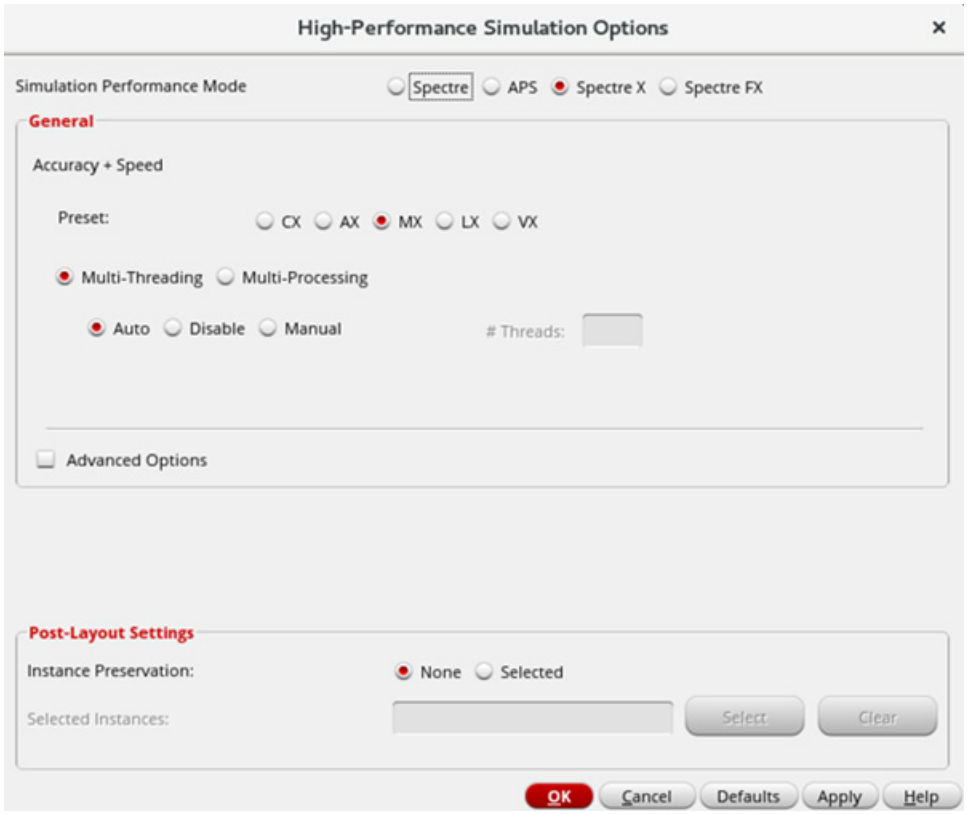
```
spectre  liberal
```

For more information on how to get better performance from APS mode, see the Getting More Performance From Spectre APS Simulation section in Spectre Classic Simulator, Spectre APS, Spectre X, and Spectre XPS User Guide.


© 2023 Cadence Design Systems, Inc.

## Spectre X

When you select the Simulation performance mode as `Spectre X`, the following fields appear in the High-Performance Simulation Options form.




**Note:** `Spectre X` can be selected as the simulation performance mode only if the AMS flex mode is enabled.

- *Preset* — Spectre X provides five high-level preset options, that allow you to trade off accuracy and performance. These high-level options not only control solver settings, but also enable parasitic optimization after automatically detecting postlayout designs.

| Option Name | When to Use |
|---|---|
| `cx` | When a golden simulation reference is needed |
| `ax` | For high-precision analog applications |
| `mx` | For most analog applications (default) |
| `lx` | For power management and other relaxed analog applications |
| `vx` | For custom IC verification |


© 2023 Cadence Design Systems, Inc.

- *Multi-Threading*: The Spectre X supports multi-threading on multi-core computer platforms. You can enable the following multi-threading modes:
  - *Auto* – To automatically allocate the threads for simulation
  - *Disable* – To disable multi-threading
  - *Manual* – To manually allocate threads
- *Multi-Processing*: Spectre X distributed simulation allows more computer cores across multiple computer hosts to further speed up DC and transient run time of large to very large postlayout circuits. You can select the following methods for job distribution: `Farm`, `SSH`, and `RSH`.

  **Note:** On selecting `Farm` as the multi-processing mode, an additional option *Sockets per dedicated Host:* is displayed. Set this option to control the number of processes allowed per node. By default, this is set to `1`. On setting a different value, the `+ppn=#` option gets added for Spectre.
- *Advanced Options:*
  - *Preset Override*: By default Spectre X ignores the user-specified simulation options that trade off performance and accuracy. However, you can select this option to tell Spectre X to honor the user-specified simulation options.The following list shows the options that are overriden by Spectre X
    - Option parameters: `iabstol`, `method`, `preserve_inst`, `reltol`, and `vabstol`
    - Transient parameters: `lteratio` and `relref`

    The following table shows the corresponding Spectre options added for the values specified in the *Preset Override* field.

| Value specified in the Preset Override field | Added Spectre Option |
|---|---|
| `reltol,method,maxstep` | `-preset_override=reltol,method,maxstep` |

**Note:** If the Preset Override option is selected but the field is left blank, Spectre honors all the user-specified simulation netlist options.


© 2023 Cadence Design Systems, Inc.

- *Parasitics Optimization*: Enables parasitic optimization with the selected preset option.
- *GPU*: Enables GPU mode of Spectre X.

  **Note:** This check box is available only when the SHELL environment variable `CDS_ADE_SPECTREX_GPU 1` is set.

  - *Auto*: Chooses the GPU devices to be used automatically if the system has multiple GPUs.
  - *Farm*: Lets the farm management tool assign the machine and GPU to be used for the simulation.

    To manually specify the number of GPUs for a simulation, specify a value in the *# Devices* field.
  - *Manual*: Lets you specify the GPUs or devices to be used to run the simulation. This enables the *# Devices IDs* field.
  - *# Devices*: Specifies the number of GPUs or devices to be used to run the simulation.
  - *# Device IDs*: Specifies the IDs for the GPUs or devices to be used to run the simulation.

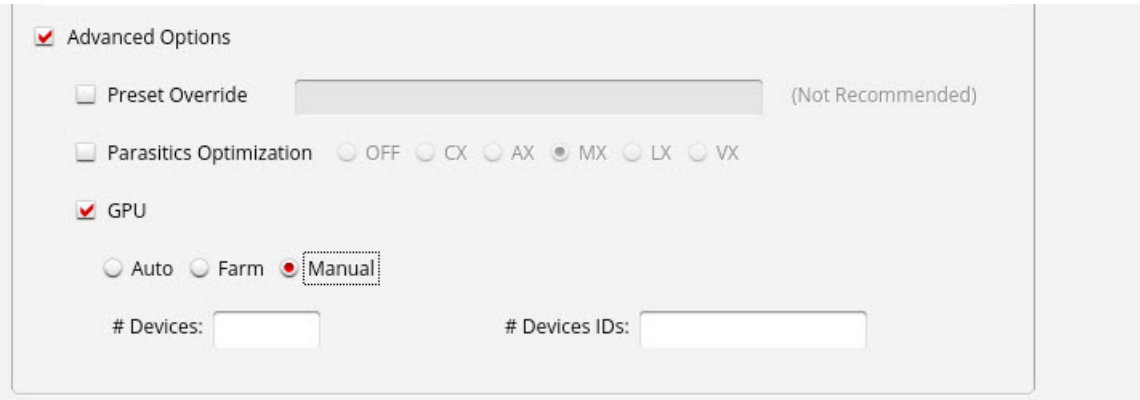
For information about Spectre X, read the *Spectre Classic Simulator, Spectre APS, Spectre X, Spectre XPS, Spectre FMC Analysis, and Legato Reliability Solution User Guide* and the *Spectre Circuit Simulator Reference*.

### Using GPU with Spectre X

Spectre X lets you use graphics processing units (GPUs) in addition to CPUs. If there are GPUs available in the setup and you have large designs that have long-running transient


© 2023 Cadence Design Systems, Inc.

analyses, post-layout or advanced node simulations, you can use GPU mode of Spectre X to speed up simulations.




To enable GPU for *Spectre X*, perform the following steps:

1. Set the SHELL environment variable `CDS_ADE_SPECTREX_GPU 1`.
2. In the High-Performance Simulation Options form, select Spectre X as the simulation performance mode.
3. Select the *Advanced Options* check box.
4. Select the *GPU* check box.

   Select a job distribution method and specify the GPUs to be used for the simulation, along with their IDs.

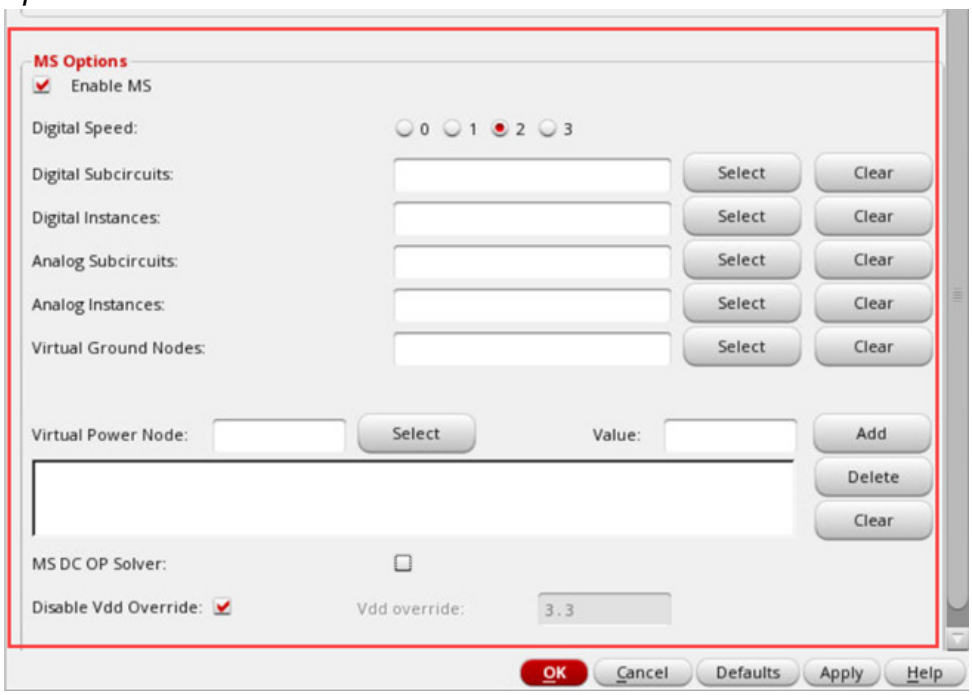
### MS Options

You can set up mixed-signal options only when:

- The selected simulation performance mode is *APS*.
- The environment variable `maestro.gui showMSOptionsInSpectreXTab` is set to `t`


© 2023 Cadence Design Systems, Inc.

On selecting *Enable MS*, the following fields appear in the *High-Performance Simulation Options* form.




**Note:** Select the *Enable Post-Layout Optimization* check box in the *Post-Layout Settings* group box if you are using MS options on a design that uses a post-layout view or post-layout files, such as DSPF files, SPEF files, or Spectre/HSPICE netlists containing post-layout parasitics.

- *Digital Speed*—Specifies the simulation speed and accuracy of digital portions. The default value is `2`.

*Tip*

Set *Digital Speed* to `1` for more accurate timing and power simulation of digital portions and set Digital Speed to `3` for faster digital verification.

- *Digital Subcircuits*—Specifies the digital subcircuits for simulation. Click *Select* to choose the digital subcircuits on the schematic.
- *Digital Instances*—Specifies the digital instances for simulation. Click *Select* to choose the digital instances on the schematic.
- *Analog Subcircuits*—Specifies the analog subcircuits for simulation. Click *Select* to choose the analog subcircuits on the schematic.
- *Analog Instances*—Specifies the analog instances for simulation. Click *Select* to choose the analog instances on the schematic.


© 2023 Cadence Design Systems, Inc.

- *Virtual Ground Nodes*—Specifies the virtual ground nodes. Click *Select* to choose the virtual ground nodes on the schematic. The value of ground voltage is assumed to be `0V`.
- *Virtual Power Node*—Specifies the virtual power supply nodes. Click *Select* to choose the virtual power supply node on the schematic.
  - *Value*—Specifies the voltage value of the virtual power supply node. After specifying the virtual power supply node and its value, click the *Add* button to add the node to the *Virtual Power Nodes* table as shown below:

**Note:** If you do not specify a voltage value for the selected virtual power supply node, the node is not added to the table when you click the *Add* button.

Select a row and click the *Delete* button to delete the selected row. Or click the *Clear* button to clear the entire table.

- *DC OP Solver*—Specifies how to calculate the DC operating point before performing the transient simulation.
  - *APS*—If selected, the DC operating point for the entire circuit is calculated by the Spectre X engine.
  - *MS*—If selected, enables the mixed-signal DC calculation.
  - *Disable Vdd Override*—Disables Vdd override. This check box is selected by default.
- *Vdd override*—Specifies the maximum voltage for the device models in the digital simulation engine. The default value is `3.3V`.

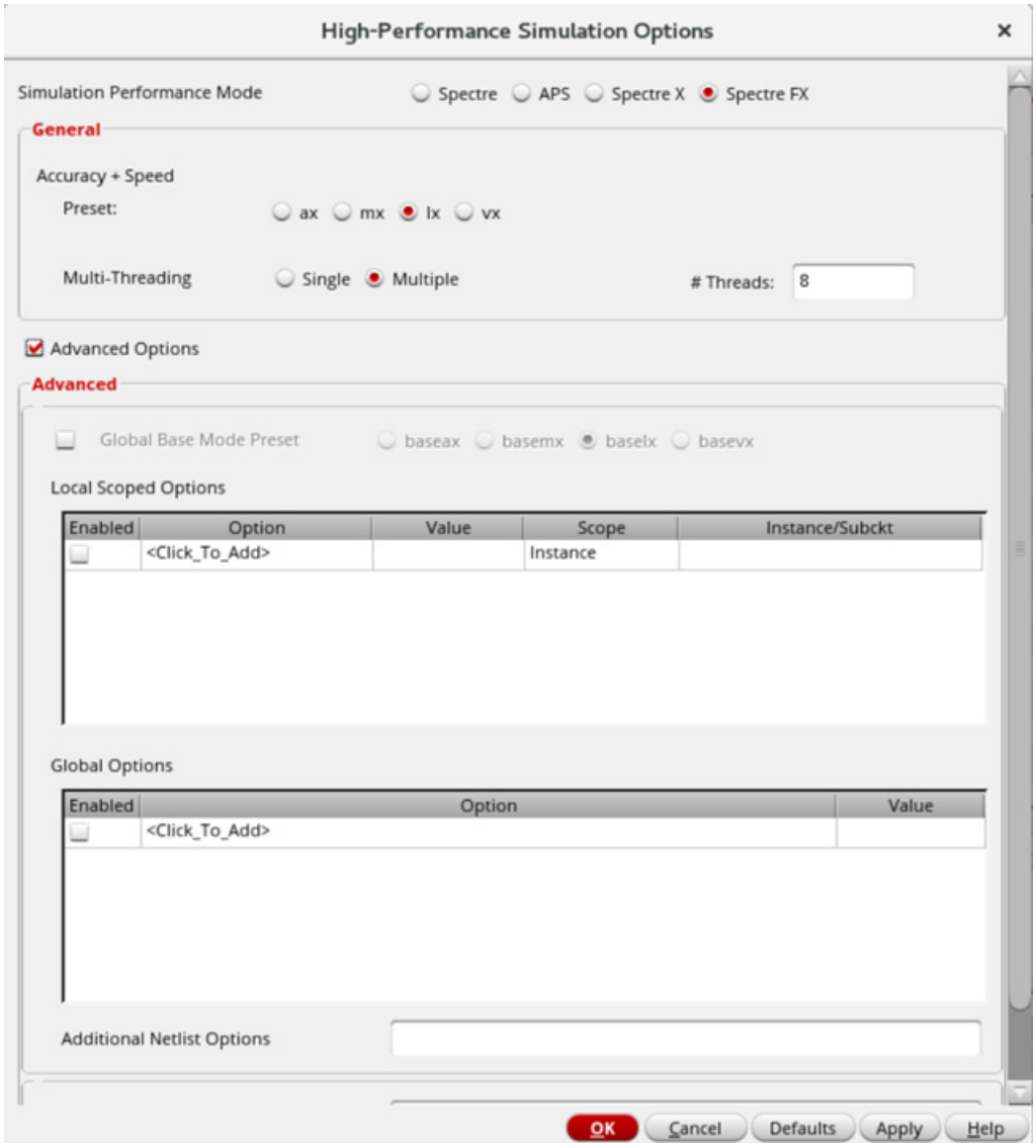
### *Related Topics*

maestro.gui showMSOptionsInSpectreXTab


© 2023 Cadence Design Systems, Inc.

## Spectre FX

On selecting the simulation performance mode as *Spectre FX*, the options specific to this mode are displayed.




In the *General* group box, specify the following options.

- *Preset* — Specifies the high-level preset options that allow you to trade off accuracy and performance. The available options are *ax*, *mx*, *lx*, and *vx*.

  By default, the *lx* preset mode is selected.


© 2023 Cadence Design Systems, Inc.

- *Multi-Threading*— Specifies the multithreading options for Spectre FX. Multithreading is supported on multi-core computer platforms.

  You can select one of the following options.

  - *Single*: uses a single thread. This is the default selection.
  - *Multiple*: lets you specify the number of threads to be used through the *#Threads* filed. The default value is `8`.

  **Note:** Spectre FX enables High Performance Computing (HPC) by default, therefore, it automatically adds multithreading jobs to the CPU cores from the same socket. This enhances the performance of multithreading.

- *Advanced Options*— Displays the *Advanced* group box.

  In this group box, specify the following options.

  - *Global Base Mode Preset*—Specifies the base mode preset mode for the entire circuit. The available options are `baseax`, `basemx`, `baselx`, and `basevx`.

    **Note:** When this check box is selected, the *Preset* option in the *Accuracy + Speed* section gets disabled.

  - *Local Scoped Options*—Specifies the options that Spectre FX lets you set locally.

    The following options can be specified using a local scope:

    - `sfx_preset`
    - `skip`

    The scope can be a subcircuit definition (`subckt=`) or instances (`inst=`).

  - *Additional Netlist Options*— Lets you specify any additional netlist options that can be used for Spectre FX.
  - *Global Options*—Lets you specify the options that Spectre FX supports. You can select an option from the drop-down list or specify other options manually.
  - *Additional Command Line Options*— Lets you specify any additional command-line options that can be used for Spectre FX.

For more information, refer to the *Spectre FX Circuit Simulator User Guide*.

### Transient Analysis with Spectre FX

When the simulation performance mode is set to Spectre FX, the Choosing Analyses form for transient analysis is updated as follows.


© 2023 Cadence Design Systems, Inc.

- The fields, *Stop Time*, *Dynamic Parameter*, and *Enabled* are available.
- On selecting the *Dynamic Parameter* check box, the options displayed in the *Param* list get updated to *User-defined*, *speed*, *maxstep*, and *temp*.

The following transient parameters are supported with Spectre FX.

- Simulation interval parameters: `start` and `outputstart`.
- Initial condition parameters: `skipdc` and `readic`
- State file parameters: `write`
- Save restart parameters: `saveclock`, `saveperiod`, `savetime`, `savefile`, and `recover`
- Strobing parameters: `skipstart`, `skipstop`, `strobeperiod`, `strobeoutput`, and `strobedelay`
- Additional parameters: `additionalParams`

### Analyses Supported by Spectre FX

Spectre FX supports the following analyses:

- It lets you run transient, DC, and AC analyses.
- It supports the *Single Run, Sweeps and Corners* and *Monte Carlo Sampling* run modes.

## Virtuoso Accelerated Parallel Simulator (APS)

*Important*

The APS interface is no longer supported because the APS options can now be set in the Spectre interface. To specify APS options, set `spectre` as the simulator and choose *Setup – High-Performance Simulation*.

Explorer provides an interface to the Virtuoso® Accelerated Parallel Simulator (APS). APS is a next generation SPICE simulator that provides high performance, high capacity circuit simulation with full Spectre accuracy. APS achieves maximum simulation performance by enabling multi-threading on multi-core and multi-CPU shared memory systems. This allows you to quickly simulate large pre- and post-layout designs.


© 2023 Cadence Design Systems, Inc.

APS combines an advanced simulation engine with existing Spectre and Spectre technologies. It is primarily targeted at speeding up DC and Transient analyses. The APS use model is identical to Spectre, with same netlist syntax, device model, analyses, features, and output format support.

For more information about APS, see the *Virtuoso Accelerated Parallel Simulator User Guide*.

## Virtuoso UltraSim Simulator Interface

Virtuoso ADE Explorer provides the interface to run Virtuoso® UltraSim™ simulator.

For more information about how to use UltraSim simulator in Explorer, see Chapter 19, "Working with UltraSim Simulator".

## Virtuoso AMS Simulator Interface

Virtuoso ADE Explorer provides a seamless integration of the Spectre AMS Designer and Xcelium Mixed-Signal simulator. The integration of the Spectre AMS Designer simulator and ADE Explorer creates a design environment with the look and feel expected by the analog and mixed-signal designers who already use ADE. When using this integration, you can access designs using the same tools you currently use for pure analog and mixed signal designs.

For more information about how to use AMS simulator in Explorer, see Chapter 15, "Working with AMS Simulator".

## Virtuoso HSPICE Simulator Interface

Virtuoso ADE Explorer provides the interface to run Virtuoso HSPICE simulator.

For more information about how to use HSPICE simulator in Explorer, Chapter 16, "Working with HSPICE Simulator".


© 2023 Cadence Design Systems, Inc.

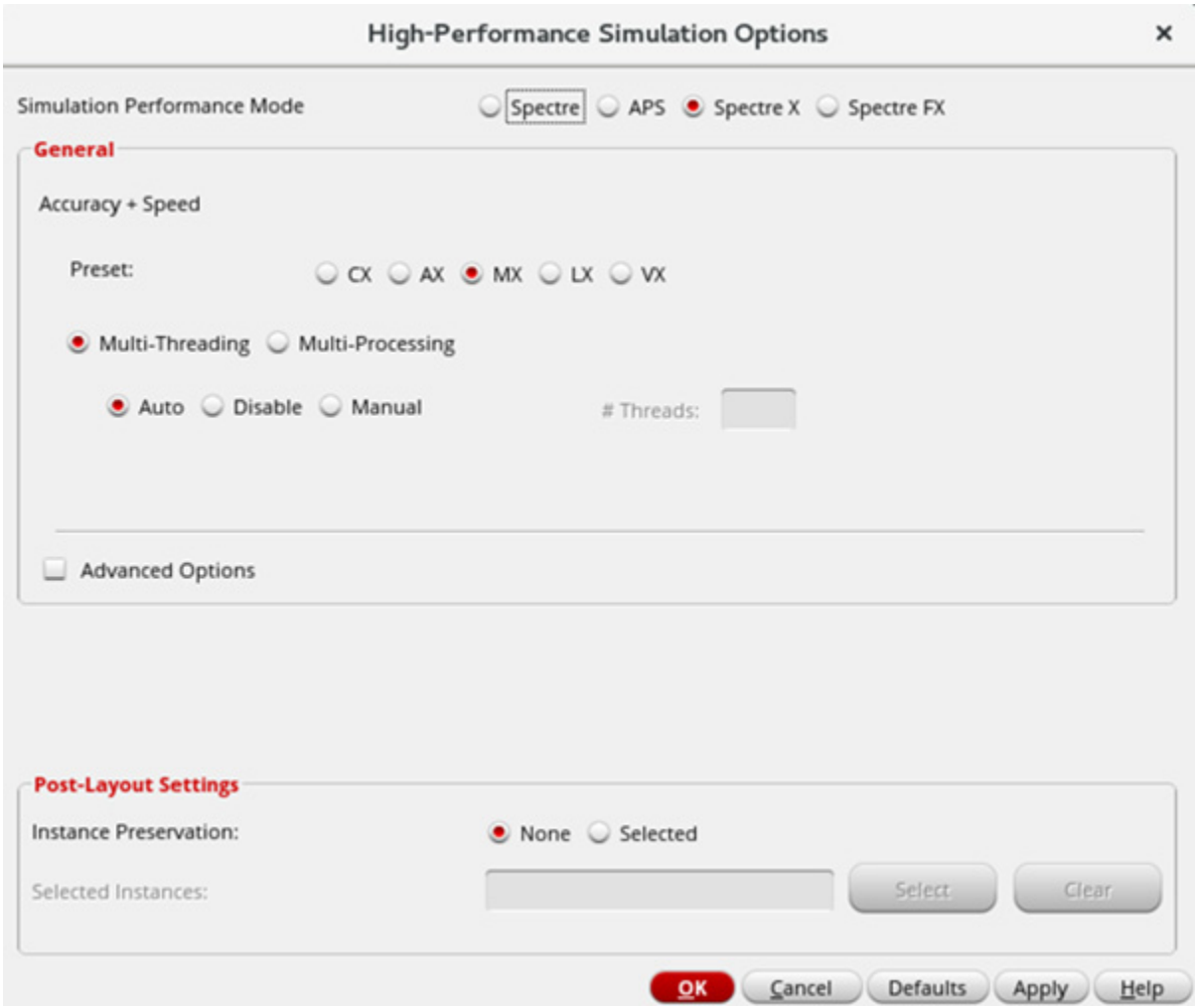
# Setting up High-Performance Simulation Options

To specify the high-performance simulation options for Spectre and AMS simulators, do the following:

1. Choose *Setup – High-Performance Simulation.*

   **Note:** The *High-Performance Simulation* menu option is disabled when a simulation is running in the interactive mode.

   The High-Performance Simulation Options form appears.




2. Select one of the following simulation performance modes:

   - *Spectre*—Use the baseline Spectre version for simulation.


© 2023 Cadence Design Systems, Inc.

- *APS*—Use the Accelerated Parallel Simulator (APS) for simulation. It preserves all of the numerical simulation algorithms like numerical tolerance control, time step control, Newton iteration control, analytical device model, and so on and it is expected to produce a near identical simulation numerical noise floor as baseline Spectre. This tight accuracy correlation between baseline Spectre and Spectre APS does restrict some additional simulation performance.

  To further speed up simulation, you can use the `++aps` simulation mode.

  You can also set up the Mixed-Signal (*MS Options*) options to get a high-performance transistor-level multi-rate simulation solution by combining a highly accurate SPICE engine (Spectre APS) together with a fast digital simulation engine (Spectre XPS). The identification between the analog and digital portions of the circuit is done automatically, with the added flexibility for advanced users to optimize it. It further speeds up both pre-layout and postlayout simulations. For more information, see MS Options.

  **Note:** To use the Mixed-Signal options during simulation, select the *Enable MS* check box.

  You can use the environment variable, uniMode to change the default setting for *Simulation Performance Mode*.

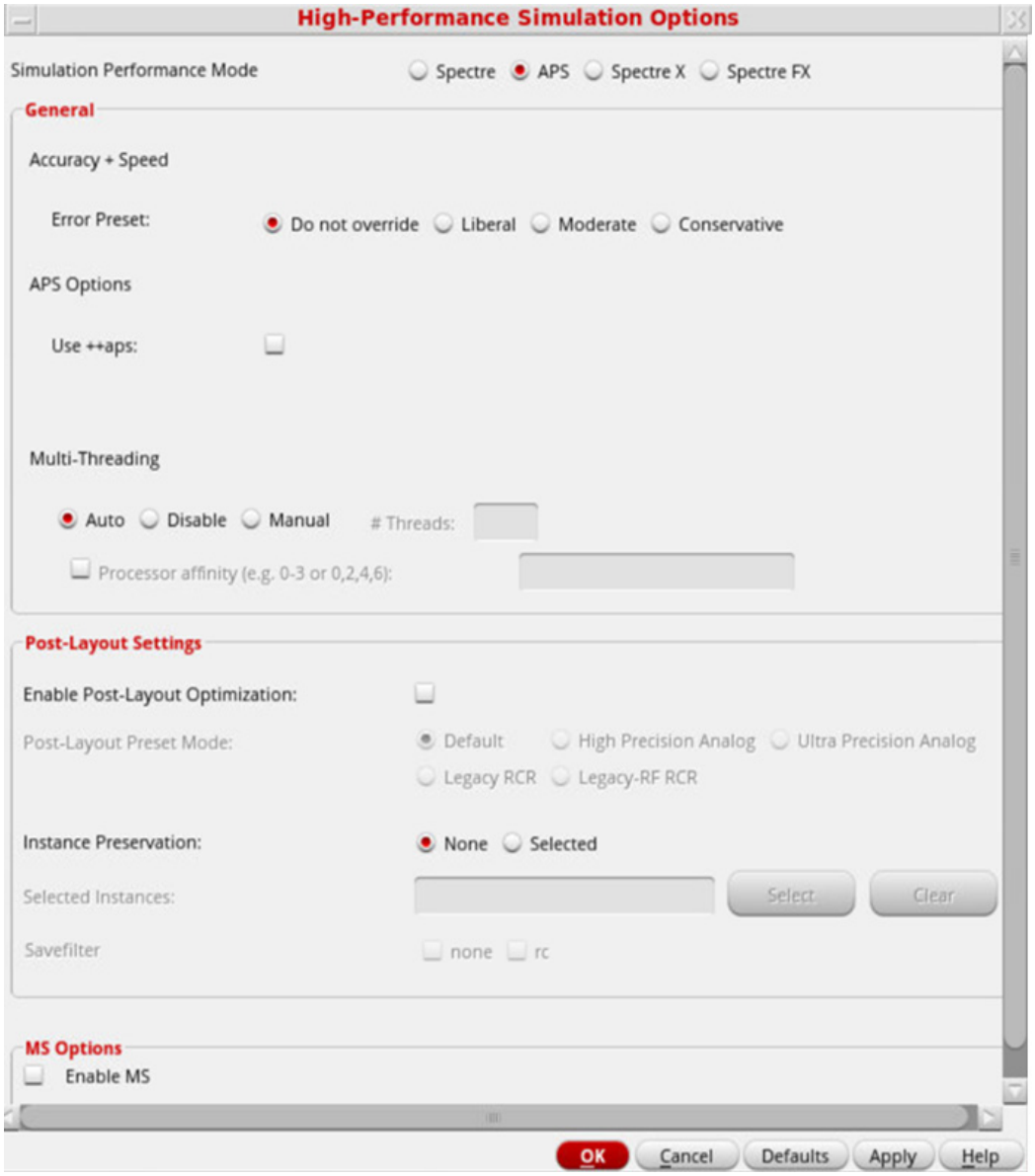
  You can switch between Spectre and APS default tolerance options using the *Reset cdsenv options* check box. When the *Simulation Performance Mode* is set to Spectre and the *Reset cdsenv options* check box is selected, ADE resets the transient options to Spectre default settings. If you want to switch to APS default settings, select APS from the *Simulation Performance Mode* and select the *Reset cdsenv options* check box.

  **Note:** The default transient options will change from Spectre to APS only if you change the *Simulation Performance Mode* from spectre to APS and select the *Reset cdsenv options* check box and vice versa. If you change the *Simulation Performance Mode* and do not select the check box, the default options will not change.

  **Note:** *Reset cdsenv options* check box is available only if the `.cdsenv` file exists in `<cds_inst_dir>/tools/dfII/etc/tools/aps/` location.


© 2023 Cadence Design Systems, Inc.

The additional fields required for APS mode appear on the form:




- *Spectre X*— Use the Spectre X simulator mode for simulation. It provides five high-level `+preset` options like `cx`, `ax`, `mx`, `lx` and `vx` that allow you to trade off accuracy and performance. These high-level options not only control solver settings, but also enable parasitic optimization after automatically detecting postlayout designs. The `cx` mode provides the highest accuracy, while the `vx` mode provides the highest performance. For more information, see Spectre X.

  **Note:** Spectre X is the default simulation performance mode.


© 2023 Cadence Design Systems, Inc.

Note the following:

- Spectre X mode is supported in Spectre 19.10 release.
- To use the Mixed-Signal options for simulation, set up the fields in the *MS Options* group box and select the *Enable MS* check box.

- *Spectre FX*— Use the Spectre® FX simulator mode for simulation. Spectre FX is the next-generation FastSPICE simulator that addresses simulation challenges like,
  - An increase in demand for performance, accuracy, and capacity from transistor-level simulators.
  - The advancement of process nodes.
  - The adoption of new design practices.
  - The need for full-chip simulation.
  - The stronger effect of parasitic elements that greatly increase the size and complexity of the designs to be simulated.

  For more information, see Spectre FX.

  **Note:** Spectre FX mode is supported in Spectre 21.10 release.

3. Specify the accuracy level for transient analysis runs by selecting one of the following options in the *Error Preset* field:
   - Select *Do not override* to use the accuracy level specified for transient analyses runs in the Choosing Analysis form. For more information about setting up a transient analysis, see Transient Analysis on page 250.
   - Select *Liberal*, *Moderate*, or *Conservative* to override the accuracy level specified for transient analyses runs in the Choosing Analyses form.

   If *errpreset* is not set for transient analyses in the Choosing Analyses form and is set to *Liberal*, *Moderate*, or *Conservative* in the High-Performance Simulation Options form, the settings are netlisted and written into the `runSimulation` file.

   For more information about the *Liberal*, *Moderate* and *Conservative* accuracy levels, see the documentation for the `errpreset` parameter in the *Spectre Circuit Simulator Reference*.

4. Select the *Use ++aps* option to enable the Fast APS mode. This mode uses a different time-step control algorithm compared to spectre, which results in improved performance while satisfying error tolerances and constraints.

   **Note:** Use ++aps option is enabled only when you set the Simulation Performance Mode


© 2023 Cadence Design Systems, Inc.

to APS or XPS.

5. Select one of the following options in the Multi-Threading field:

| Select | To |
|---|---|
| *Auto* | Use the maximum number of available threads to run the simulation.<br><br>The number of threads used—one thread for every CPU core—is automatically derived from the hardware architecture in use, but is limited to a maximum of:<br><br>■ 4 threads when Spectre is set as the simulation performance mode<br><br>■ 32 threads when APS is set as the simulation performance mode. |
| *Manual* | When a manual multi-threading simulation is run, in the *#Threads* field, you can add either:<br><br>■ The number of threads. You can specify up to:<br><br>❑ 4 threads when Spectre is set as the simulation performance mode<br><br>❑ 64 threads when APS is set as the simulation performance mode<br><br>**Note:** The tooltip of #Threads field shows the estimated thread count, which is the CPU estimate value backannotated from the Job Policy Setup form<br><br>■ The computer farm management tool (`sge`, `lsf`, `loadleveler` or `farm`) to be used to run simulation. On doing so, a `+mt` option will be added to the `runSimulation` file in the netlist directory.<br><br>For example, if the specified computer farm management tool is `sge`, the option, `+mt=sge` will be added to the `runSimulation` file.<br><br>**Note:** In the manual multi-threading mode, the environment variable spectreMTNumberUsed specifies the `+mt` option for Spectre APS. |
| *Disable* | Disable multithreading. |

Note the following:


© 2023 Cadence Design Systems, Inc.

- The multithreading options specified in this form override the multithreading options specified in the Simulator Options form. For more information about the Simulator Options form, see Specifying Simulator Options on page 102.
- If *Spectre* is set as the analog solver for the Spectre AMS Designer simulator, multithreading is supported only if you are using the Cadence IUS 8.2 or a later release.

**6.** Select one of the following options in *Savefilter*:

| Select | To |
|---|---|
| *none* | Speeds up simulation by including RC reduction. |
| *rc* | Enables filtering the internal RC nodes for probe or save statement. It is the default value of *Savefilter*. |

**7.** In the Processor affinity field, specify the processors using which simulation sessions should be run.

*Important*

This field is available only for the lnx86 platform, If you are using an ibmrs or a sun4v machine, this option will not be visible.

When running multiple multi-threading simulation sessions on a single computer, it is recommended that you fix the simulation session to particular processors. Otherwise multiple simulation sessions can race against each other to get the available processors, resulting in a less than optimum simulation performance. For example, to run two 4-thread simulation sessions on an 8-processor computer, specify 0-3 in the Processor affinity field for the first 4-thread simulation session using processors 0 to 3, and 4-7 in the Processor affinity field for the second 4-thread simulation session using processors 4 to 7.

**Note:** You can specify a range of processors such as 0-4, or a comma-separated list of processors such as `0,2,4,6`.

**8.** To optimize post-layout designs and maximize post-layout simulation performance select the *Enable Post-Layout Optimization* check box.

**Note:** The *Enable Post-Layout Optimization* check box and other options in the *Post-Layout Settings* section are available only when you set the Simulation Performance Mode to APS or XPS MS.


© 2023 Cadence Design Systems, Inc.

**9.** Select one of the following options in the *Post-Layout Preset Mode* field:

| Select | To |
|---|---|
| *High Precision Analog* | Perform post-layout optimization with better accuracy, as compared to the default mode, but with slower performance. You can use this option for sensitive analog and mixedsignal designs.<br><br>This is equivalent to the `+postlayout=hpa` Spectre command-line option. |
| *Ultra Precision* | Perform post-layout optimization with the best accuracy and include the optimizations present in the High Precision Analog mode. |
| *Analog* | This is equivalent to the `+postlayout=upa` Spectre command-line option |
| *Legacy RCR* | This is equivalent to the `+postlayout=legacy` Spectre command-line option, which is similar to the `+parasitics` Spectre command-line option. |
| *Legacy-RF RCR* | Perform RF analysis. It preserves the level of accuracy needed for RF analysis.<br><br>This is equivalent to the *+postlayout=legacy_rf* Spectre command-line option, which is similar to the *+parasitics=rf* Spectre command-line option. |

**Note:** This field appears only when the performance mode is set to APS.

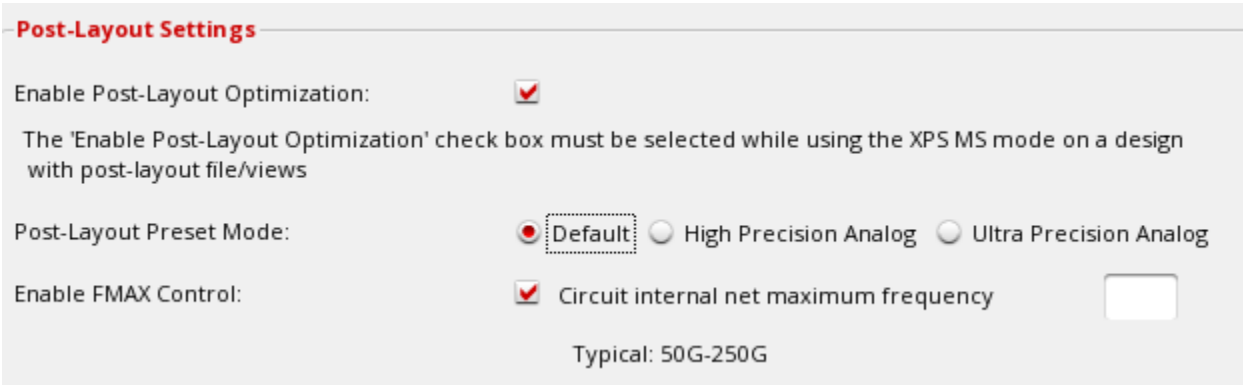
**10.** Select the *Enable FMAX Control* check box to specify the maximum frequency of the internal net of the circuit or the max operating frequency of the circuit.

**Note:** This field appears only when the XPS MS simulation performance mode is selected and the *Enable Post-Layout Optimization* check box is selected.

- *Circuit internal net maximum frequency*—Specify an integer value in this field. It adds the `+rcnet_fmax=<value>` option to the `+postlayout` command line option. The maximum frequency typically ranges from 50G to 250G.
  If the simulation performance mode is set to APS, this field appears when the post-layout preset mode is set to *Default*, *High Precision Analog*, or *Ultra Precision*

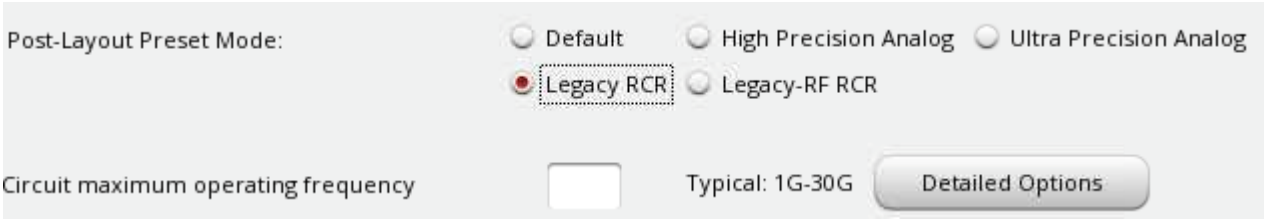

© 2023 Cadence Design Systems, Inc.

*Analog*. And if the performance mode is set to XPS MS, this field appears when the *Enable FMAX Control* check box is selected.




- *Circuit maximum operating frequency*—Specify an integer value in this field. It adds the `rcr_fmax=<value>` global option to the netlist. The operating frequency typically ranges from 1G to 30G.

  This field appears only when the performance mode is set to APS and the post-layout preset mode is set to *Legacy RCR*.



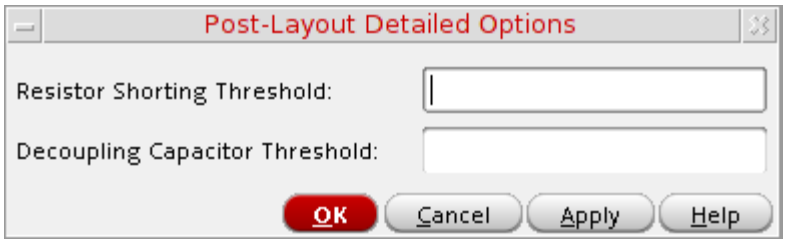

**11.** Specify the instances that you do not want to be shorted by selecting the required *Instance Preservation* option:

| Select | To |
|---|---|
| *None* | Specify that no instances need to be preserved from being shorted. |


© 2023 Cadence Design Systems, Inc.

| Select | To |
|---|---|
| *Selected* | Specify the instances that must be preserved from being shorted.<br><br>In the *Preserve Instance* field, type the instance names separated by spaces. You can also use wildcards to specify the instances. For example, type `I0.I1.r*` to specify that all instances with names starting with `r` in the `I0.I1` hierarchy must be preserved from being shorted.<br><br>To select instances from the schematic, do the following:<br><br>**1.** Click the *Select* button next to the *Preserve Instance* field.<br><br>The schematic for the design is displayed.<br><br>**2.** Select one or more instances on the schematic.<br><br>To select more than one instance at a time, do one of the following:<br><br>■ Hold down the *Shift* key and click the instances you want to select.<br><br>■ Drag the mouse pointer over the instances you want to select.<br><br>All the instances that are within the yellow bounding box that appears are included in the selection.<br><br>**3.** Press the *Esc* key when you are done.<br><br>The selected instances are displayed in the *Preserve Instance* field. |
| All | Specify that all instances should be preserved from being shorted. |

**12.** (Optional) Click the *Detailed Options* button. The Post-Layout Detailed Options form appears:

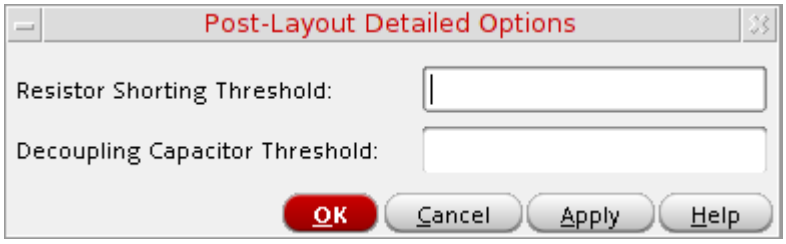



In the form, specify the following:

❑ *Resistor Shorting Threshold*— Specify a threshold value (in `ohms`) to short resistors whose resistance is less than the specified value.


© 2023 Cadence Design Systems, Inc.

**13.** (Optional) Click the *Detailed Options* button. The Post-Layout Detailed Options form appears:




In the form, specify the following:

- *Resistor Shorting Threshold*— Specify a threshold value (in `ohms`) to short resistors whose resistance is less than the specified value.
- *Decoupling Capacitor Threshold*—Specify a threshold value (in `farads`) to decouple capacitors whose capacitance is less than the specified value.

The *Detailed Options* button is enabled only when APS is selected as the performance mode and the *Enable Post-Layout Optimization* check box is selected.

**14.** Click *OK*.

**Note:** If high-performance mode is enabled, the simulator mode is automatically changed to `batch`.

The ADE Explorer window displays the settings you made in the High-Performance Simulation Options form. For example, if you selected the simulation performance mode as and the accuracy level as *Liberal*, the status bar in the window displays the simulator name as:

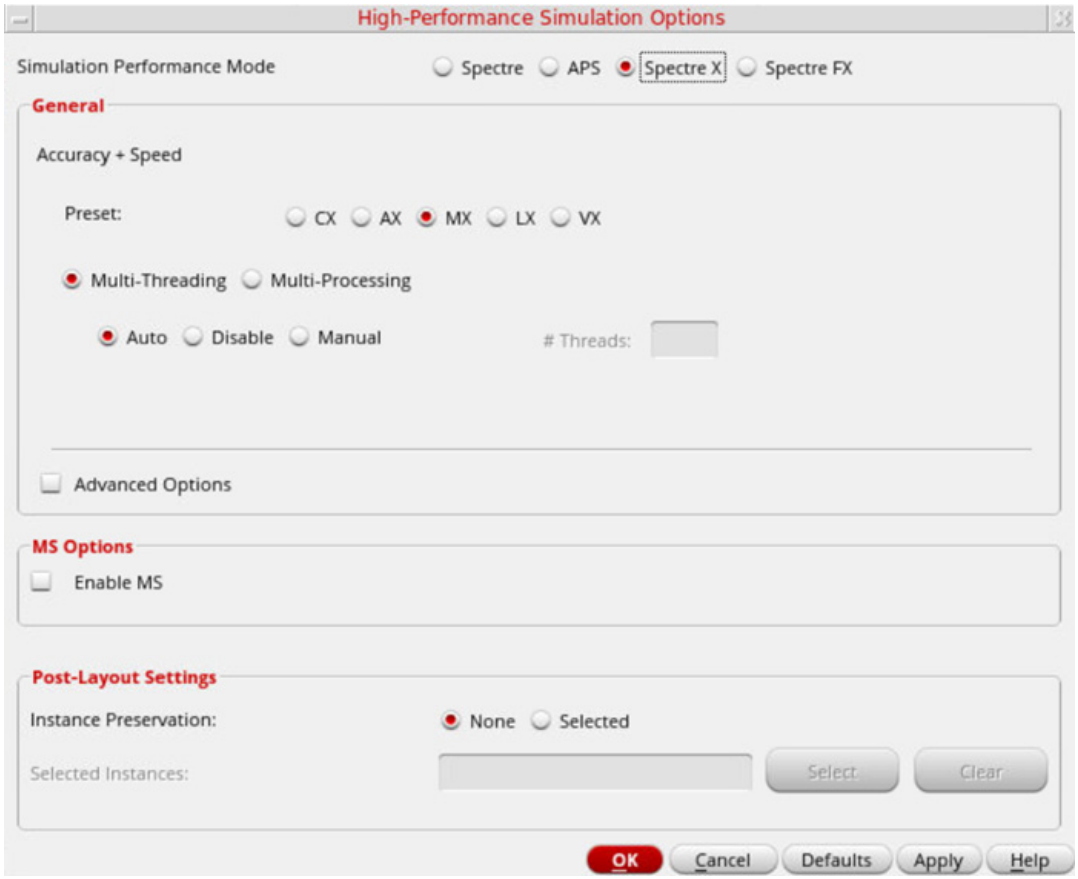
```
spectre  liberal
```

For more information on how to get better performance from APS mode, see the Getting More Performance From Spectre APS Simulation section in Spectre Classic Simulator, Spectre APS, Spectre X, and Spectre XPS User Guide.


© 2023 Cadence Design Systems, Inc.

### Spectre X

When you select the Simulation performance mode as `Spectre X`, the following fields appear in the High-Performance Simulation Options form.




**Note:** `Spectre X` can only be selected as the *Simulation Performance Mode* if the AMSD flex mode is enabled.

- *Preset* — Spectre X provides five high-level preset options, that allow you to trade off accuracy and performance. These high-level options not only control solver settings, but also enable parasitic optimization after automatically detecting postlayout designs.

| Option Name | When to Use |
|---|---|
| cx | When a golden simulation reference is needed |
| ax | For high-precision analog applications |
| mx | For most analog applications (default) |


© 2023 Cadence Design Systems, Inc.

| Option Name | When to Use |
|---|---|
| lx | For power management and other relaxed analog applications |
| vx | For custom IC verification |

Related environment variable: spectreXPreset

- *Multi-Threading* — The Spectre X supports multi-threading on multi-core computer platforms. You can enable the following multi-threading modes:
  - *Auto* – To automatically allocate the threads for simulation
  - *Disable* – To disable multi-threading
  - *Manual* – To manually allocate threads

  Related environment variable: spectreXMtOption
- *Multi-Processing* — Spectre X distributed simulation allows more computer cores across multiple computer hosts to further speed up DC and transient run time of large to very large postlayout circuits. You can select the following methods for job distribution: `Farm, SSH,` and `RSH`.

  Related environment variable: spectreXMpOption
- *Advanced Options* —
  - *Preset Override* – By default Spectre X ignores the user-specified simulation options that trade off performance and accuracy. However, you can select this option to tell Spectre X to honor the user-specified simulation options.
  - *Parasitics Optimization* – Select the check-box to enable Parasitics optimization with the selected preset option.

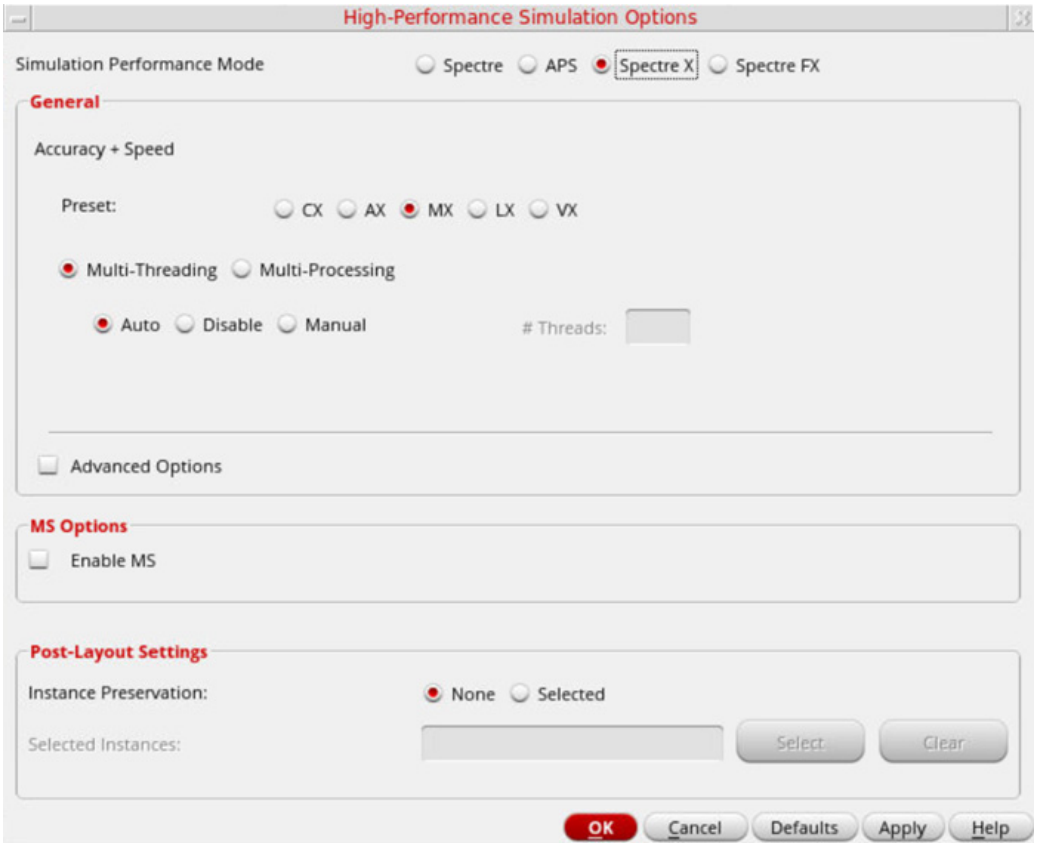
  Related environment variable: spectreXAdvanced
- *MS Options* — Set up the mixed-signal options to get a high-performance transistor-level multi-rate simulation solution using Spectre X engine. The identification between the analog and digital portions of the circuit is done automatically, with the added flexibility for advanced users to optimize it. It further speeds up both pre and post-layout simulations. For more information, see MS Options.

For information, see the Spectre X environment variables section, and read the *Spectre Circuit Simulator User Guide* and the *Spectre Circuit Simulator Reference*.


© 2023 Cadence Design Systems, Inc.

## MS Options

You can only set up the Mixed-Signal options when the simulation performance mode is set to *APS* or *Spectre X*. On selecting the *Enable MS* check box, the following fields appear in the *High-Performance Simulation Options* form.




**Note:** You need to select the *Enable Post-Layout Optimization* check box if you are using the MS options on a design that uses a post-layout view or post-layout files, such as DSPF files, SPEF files, or Spectre/HSPICE netlists containing post-layout parasitics.

This form includes an additional section, *MS Options*, that includes the various MS options.


© 2023 Cadence Design Systems, Inc.

The options that appear in the *MS Options* section of the High-Performance Simulation Options form are described below:

- *Digital Speed*—Specifies the simulation speed and accuracy of digital portions. The default value is `2`.

*Tip*

Set *Digital Speed* to `1` for more accurate timing and power simulation of digital portions and set Digital Speed to `3` for faster digital verification.

- *Digital Subcircuits*—Specifies the digital subcircuits for simulation. Click *Select* to choose the digital subcircuits on the schematic.
- *Digital Instances*—Specifies the digital instances for simulation. Click *Select* to choose the digital instances on the schematic.
- *Analog Subcircuits*—Specifies the analog subcircuits for simulation. Click *Select* to choose the analog subcircuits on the schematic.
- *Analog Instances*—Specifies the analog instances for simulation. Click *Select* to choose the analog instances on the schematic.
- *Virtual Ground Nodes*—Specifies the virtual ground nodes. Click *Select* to choose the virtual ground nodes on the schematic. The value of ground voltage is assumed to be `0V`.
- *Virtual Power Node*—Specifies the virtual power supply nodes. Click *Select* to choose the virtual power supply node on the schematic.
  - *Value*—Specifies the voltage value of the virtual power supply node. After specifying the virtual power supply node and its value, click the *Add* button to add the node to the *Virtual Power Nodes* table as shown below:

**Note:** If you do not specify a voltage value for the selected virtual power supply node, the node is not added to the table when you click the *Add* button.

Select a row and click the *Delete* button to delete the selected row. Or click the *Clear* button to clear the entire table.


© 2023 Cadence Design Systems, Inc.

- *DC OP Solver*—Specifies how to calculate the DC operating point before performing the transient simulation.
  - *APS*—If selected, the DC operating point for the entire circuit is calculated by the Spectre X engine.
  - *MS*—If selected, enables the mixed-signal DC calculation.
  - *Disable Vdd Override*—Disables Vdd override. This check box is selected by default.
- *Vdd override*—Specifies the maximum voltage for the device models in the digital simulation engine. The default value is `3.3V`.

For information about Spectre, read the *Spectre Circuit Simulator User Guide* and the *Spectre Circuit Simulator Reference*.

## Virtuoso Accelerated Parallel Simulator (APS)

*Important*

The APS interface is no longer supported because the APS options can now be set in the Spectre interface. To specify APS options, set `spectre` as the simulator and choose *Setup – High-Performance Simulation*.

Explorer provides an interface to the Virtuoso® Accelerated Parallel Simulator (APS). APS is a next generation SPICE simulator that provides high performance, high capacity circuit simulation with full Spectre accuracy. APS achieves maximum simulation performance by enabling multi-threading on multi-core and multi-CPU shared memory systems. This allows you to quickly simulate large pre- and post-layout designs.

APS combines an advanced simulation engine with existing Spectre and Spectre technologies. It is primarily targeted at speeding up DC and Transient analyses. The APS use model is identical to Spectre, with same netlist syntax, device model, analyses, features, and output format support.

For more information about APS, see the *Virtuoso Accelerated Parallel Simulator User Guide*.


© 2023 Cadence Design Systems, Inc.

# Setting Up Simulation Files

Before you run a simulation, you must set up the simulation files you want to use.

➤ Choose *Setup – Simulation Files*.

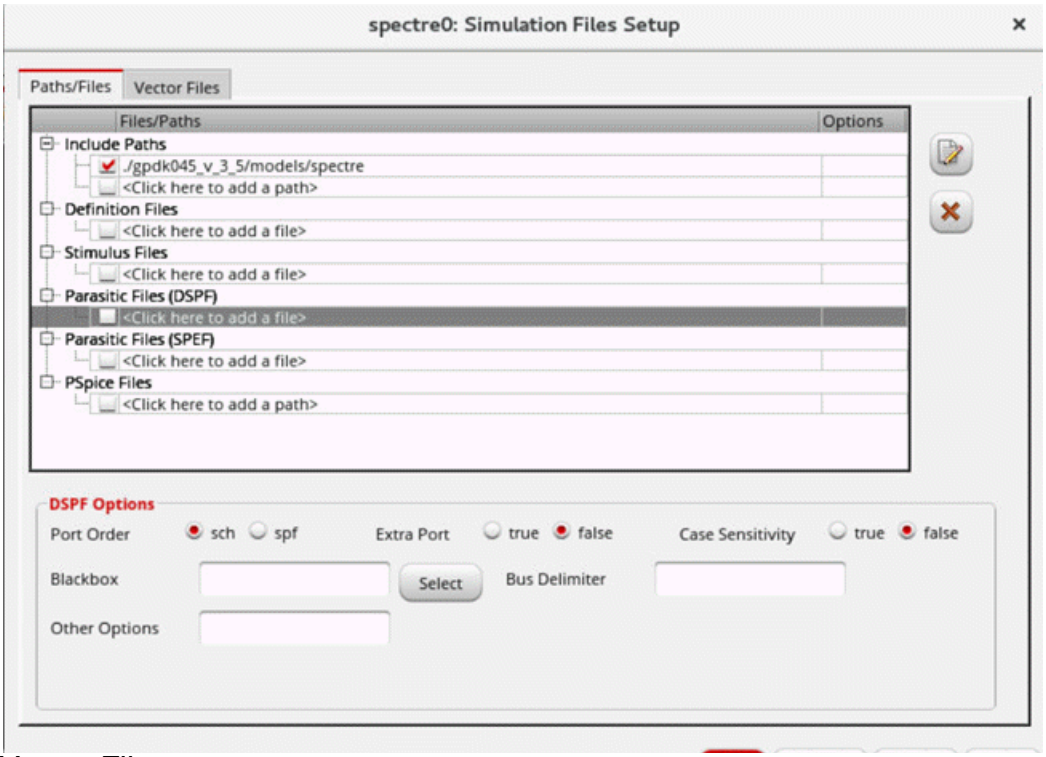
The Simulation Files Setup form appears.

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.

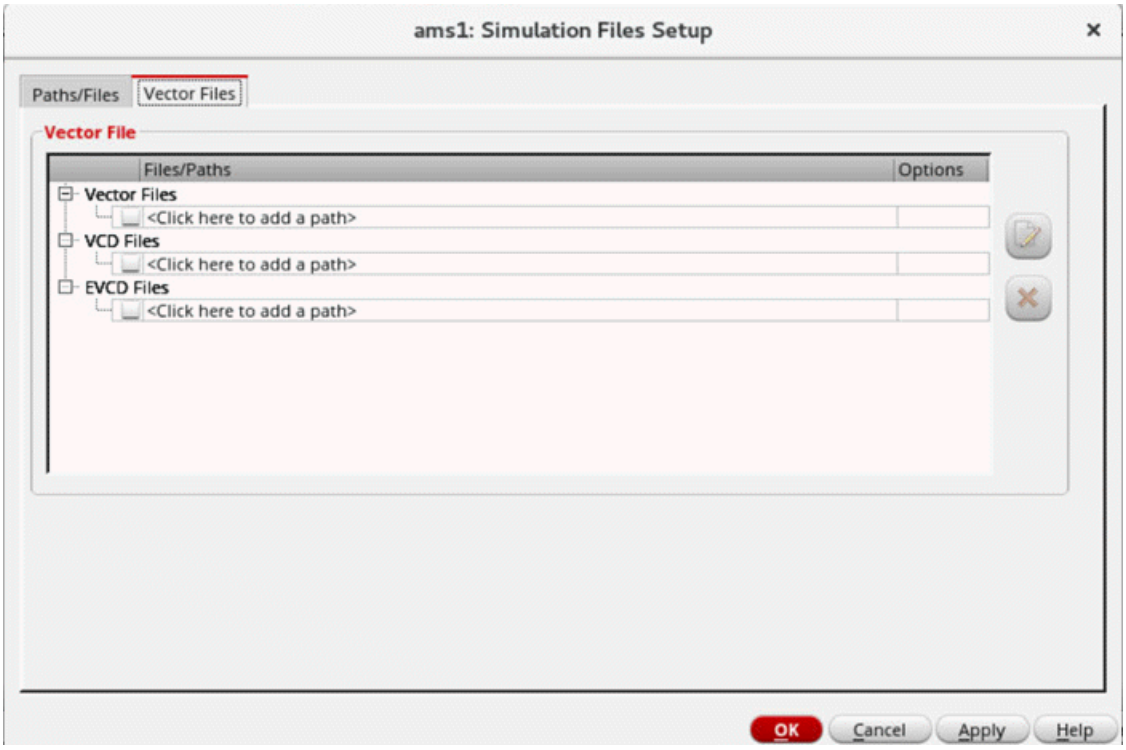
**Note:** The fields that appear on the form depend on your target simulator.


© 2023 Cadence Design Systems, Inc.

Path/Files:




Vector Files:





© 2023 Cadence Design Systems, Inc.

Refer to the following sections for more information about using the *Simulation Files Setup* form:

- Specifying Simulation Files on page 157
- Specifying Simulation Files Using Variables on page 160
- Using Shell Environment Variables to Specify the Paths on page 161
- Enabling and Disabling Simulation Files on page 162
- Editing Simulation Files on page 162

## Specifying Simulation Files

Depending on your design and setup requirements, you need to use various types of files for simulations run in ADE Explorer.

The table below describes each file type and its purpose.

| File Type | Description |
|---|---|
| Definition Files | Definition files contain function and parameter definitions that are not displayed in the *Design Variables* list in the Data View assistant. See the following sample file that contains function and parameter definitions for the Spectre circuit simulator.<br>`<your_install_dir>/tools/dfII/samples/artist/models/spectre/definitions.scs`<br>The parameters in this file are referenced by included models and are not referenced from any part of the design in the Cadence library. |


© 2023 Cadence Design Systems, Inc.

| | |
|---|---|
| Stimulus Files | Stimulus files can contain input and power supply stimuli, initializing nodes, and 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.<br>`<your_install_dir>/tools/dfII/samples/artist/models/spectre/opampStimuli.scs`<br>In a 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*. |
| DSPF Parasitic Files | DSPF files contain the post-layout data, including fingered devices and net parasitics, of the design. For EMIR analysis, Spectre APS/XPS requires a complete testbench that contains the DSPF files (with the parasitic and instance sections describing the circuit to be analyzed), stimuli, and device models.<br>On specifying a DSPF parasitic file, a few additional options are displayed. For more information on these options, see Specifying DSPF Files in the Simulation Files Setup Form in the *Virtuoso ADE Assembler User Guide*. |
| SPEF Parasitic Files | SPEF post-layout netlists are supported by the AMS, Spectre APS/XPS EMIR flow. An SPEF netlist contains a combination of a schematic netlist with the active devices and an SPEF netlist that includes the parasitic element information. |
| PSpice® Files | PSpice files contain netlists that include PCB components modeled using the PSpice format.<br>When you use PSpice files, the top-level netlist and control statement need to be defined in Spectre or SPICE format. A subcircuit defined in PSpice netlist format is instantiated in the Spectre netlist.<br>**Note:** For the AMS simulator, the *PSpice Files* field is supported only when you select AMS Unified Netlister with xrun from the Netlist and Run Options form. For information on AMS UNL, see Working with AMS Simulator. |
| Vector Files | Vector (VEC) files include a table of times and values at those times. A vector file can include more than one vector definition. |


© 2023 Cadence Design Systems, Inc.

| | |
|---|---|
| VCD Files | VCD (Verilog® Value Change Dump) files contain information about value changes for selected variables in the circuit design. |
| EVCD Files | EVCD (Extended Verilog® Value Change Dump) files contain variable changes in all states and provide strength information and port direction. |

You can use the Simulation Files Setup form to specify the paths to these files.

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

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

   The Simulation Files Setup form appears. This form contains two tabs:

   - *Paths/Files*: On this tab, you can specify paths for all file types other than vector files.
   - *Vector Files*: On this tab, you can specify digital vector files.

2. If the files to be used are located at multiple locations and you want ADE Explorer to look for files in all those paths, specify those paths in the *Include Paths* tree.
3. To specify the path to a file of a specific type, click where it says *<Click here to add a path>* or *<Click here to add a file>* in the tree of that file type.

   - For example, to specify the path to a definition file, click where it says *<Click here to add a file>* in the *Definition Files* tree.

4. Type the path and name of your file, or click the browse button to select one or more files or paths using the Choose Files/Paths form.

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

   After you have specified a file or path, a new row is added to the tree. If you have multiple files, you can use the additional rows to specify the paths to those files.

5. (Optional) Click *Apply to the Selected Tests in the Setup* () on the *Paths/Files* tab of this form to update simulation files for all the applicable tests in a setup.

   The Apply Simulation Files Setup to Tests form appears.

   a. Select the tests to which you want to apply the simulation files.


© 2023 Cadence Design Systems, Inc.

You can also click *Select Enabled* to automatically select the tests that are enabled in the Data View assistant.

**b.** Click *OK* to apply the selected simulation files.

***When you apply the simulation files to other tests, their existing simulation files setup is overwritten. Therefore, it is recommended to verify their existing setup before using this feature.***

**6.** Click *OK* to close the Simulation Files Setup form.

***Important Points to Note***

- For AMS simulator, a *PSpice_include Statements included in Model Files* check box appears in the *PSpice Options* section of the Simulation Files Setup form
- For some of the file types, a few additional fields are displayed on the form.

  For example, when you select a vector file, the *Vector File Options* section is displayed on the *Vector Files* tab with the *hlCheck* field. To enable the check for H and L states for input signals, choose *1.*

  Selecting a VCD or EVCD file displays an additional field in which you can specify a VCD or EVCD file is.
- By default, the simulation files you add are enabled for simulation. You can use the check box next to a file name to enable or disable a simulation file for simulation.
- To edit a simulation file, select the file and click *Edit/view selected files*. The selected simulation file is opened in a text editor where you can make changes to the file content and save it.
- To remove a simulation file from the setup, select it and click *Delete selected rows*.

## Specifying Simulation Files Using Variables

You can use variables to specify simulation files. Variables allow you to vary the simulation files during simulation.

For example, to specify a definition file using a variable, 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>.*

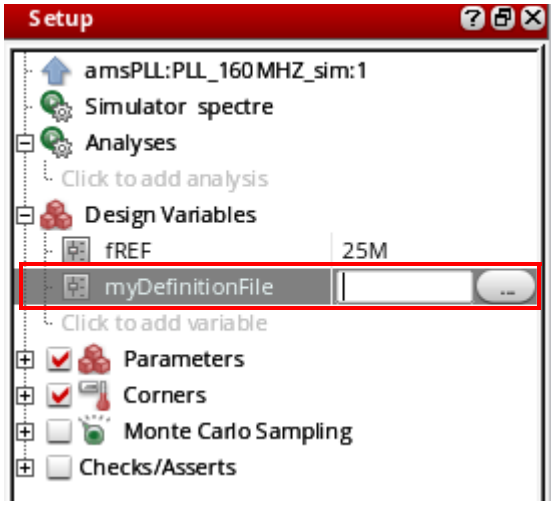

© 2023 Cadence Design Systems, Inc.

3. Type a variable name for the definition file you want to vary using the following format:

   ```
   VAR("myDefinitionFile")
   ```

4. Click *Apply*.

   The `myDefinitionFile` variable appears in the *Design Variables* section of the Setup assistant.




5. Double-click in the field next to the variable name and type the path and filename of the stimulus file or select the file using the browse button.

## Using Shell Environment Variables to Specify the Paths

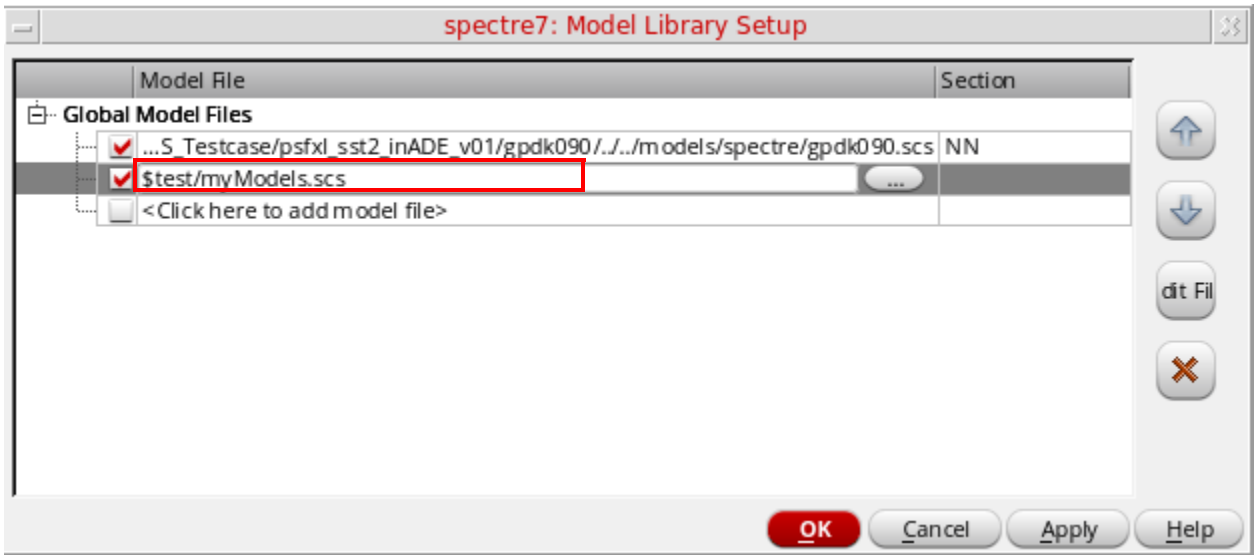
You can use shell environment variables to specify the paths of the simulation, vector and model files in the simulation window. Shell environment variables allow you to set different file paths for each session.

For example, to specify a model file using the shell environment variable, do the following:

1. In the *Global Model Files* tree, click the cell with text *<Click here to add model file>.*
2. Type the name of the shell environment variable to include the model file path in the following format:


© 2023 Cadence Design Systems, Inc.

```
$<shell_variable_name>/<model_file_name>
```




3. Click *Apply*.

   **Note:** The shell environment variables are not displayed in the *Design Variables* section of the Setup assistant. The paths from the shell environment variables are resolved internally during the simulation run.

## 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.


© 2023 Cadence Design Systems, Inc.

*Important*

If you edit simulation files outside ADE Explorer, the changes in the files will not be included in the netlist if you click the *Netlist and Run* button or choose *Simulation – Netlist and Run*. You must choose *Simulation – Netlist – Recreate* to include the changes in the netlist.

## Deleting Simulation Files

To delete simulation files, do the following:

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

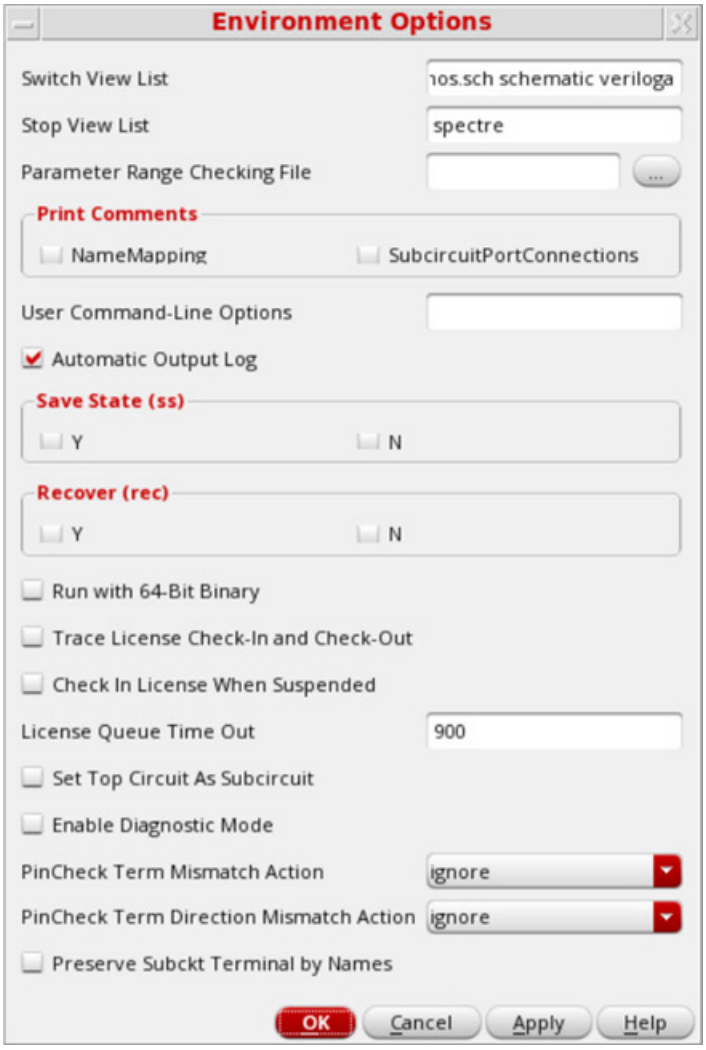

© 2023 Cadence Design Systems, Inc.

# Setting up Simulator Environment Options

You can set the environment options for the specified simulator using the Environment Options form. To open the Environment Options form:

1. Choose *Setup – Environment*.

   The *Environment Options* form appears.




   The display varies depending on which simulator you are using and whether you are using a config view for the design. Instance-based view switching is supported only for purely analog designs, not for mixed-signal designs.

2. Check that the path to the parameter range-checking file is correct. For the Spectre simulator, this file contains the parameter range limits. You do not need to enter the full


© 2023 Cadence Design Systems, Inc.

path for the file if the file is in the directory specified in the include path on the Model Setup form.

**Note:** A period (.) in a UNIX path specification is interpreted relative to the directory from which you started the ADE Explorer.

3. (Optional) If you are not using a config view, check and set the options for view switching to control how the system netlists hierarchical designs.
4. Set the other options as listed below, and click *OK*.

   - *Switch View List*—This is a list of the views that the software switches into when searching for design variables.—The software searches through the hierarchical views in the order shown in the list.
   - *Stop View List*—This is a list of views that identify the stopping view to be netlisted. This list does not require a particular sequence.
   - *Parameter Range-Checking File*—Lets you enter the path to a file containing the correct ranges for component parameters. If this path is present, the simulator checks the values of all component parameters in the circuit against the parameter range-checking file and prints out a warning if any parameter value is out of range.
   - *Print Comments*
     - Prints the name mapping of the terminals in the netlist as comments.
     - When *Name Mapping* is selected, prints the mapping of the schematic terminal name with the netlist terminal name for the subcircuits and the mapping of schematic device names with simulator devices names.
     - When *Subckt Port Connections* is selected, prints the connection of each terminal with the net it is connected to, for each subcircuit.
   - *User Command-line Options*—Helps you specify options that cannot otherwise be specified using the UI. You can enter commands that are valid for the selected simulator. ADE appends these commands as they are to the list of commands specified using the UI for the simulator. If you provide invalid commands through this option, ADE does not validate them; the simulator may or may not fail depending upon the simulator strategy applied to invalid commands.

     To enable the MTS flow in Spectre, specify `+mts` command in this field.
   - *Automatic output log*—When this check box is selected, the output log opens and displays simulator messages as they are generated.

     **Note:** The output log appears only for the single-point simulation runs. If the simualtion run includes corners or sweeps, the output log is not displayed


© 2023 Cadence Design Systems, Inc.

regardliess of the setting of the *Automatic output log* check box.

- *savestate (ss)*
  - **Y**—Runs spectre with the `+ss` option, which saves circuit information at set intervals or at multiple points during a transient analysis.
  - **N**—Runs spectre with the `-ss` option, which does not save circuit information at set intervals during a transient analysis.

  **Note:** This option does not work as intended with parametric analysis. Ensure that it is set to N, it is the default setting, before using these tools.

- *recover (rec)*—Allows the simulation to be restarted from a specified checkpoint.
  - **Y**—Runs spectre with the `+rec` option, which restarts a transient simulation based on conditions specified in a saved-state file.
  - **N**—Runs spectre with the `-rec` option, which does not restart a transient simulation, even if conditions for this have been specified in a saved-state file.

  *Run with 64 bit binary*—Instructs spectre to run in 64 bit mode.

- *Using colon as Term Delimiter*—Instructs spectre to use : as the terminal delimiter. Whenever you select sst2 as the output format, you have to explicitly select this form option.

- *Netlist text views from shadow OA database*—While generating the netlist for a design that contains text cellViews,if this option is selected, the netlister reads the database for each text cellview and prints the corresponding netlist.

  If this option is not selected, the netlister includes the corresponding text files in the netlist.

  **Important Points to Note:**
  - This form option will have an impact only when you select `sst2` as the output format.
  - If *Using colon as Term Delimiter* option is selected/deselected, you need to netlist the design.
  - *Trace License Check-in/Check-out*—Writes the license check-in/check-out information in the log file.

- *Check in license when suspended*—Turns on the license suspend/resume capability. When Spectre receives `SIGTSTP`, it checks in all the licenses before it gets suspended. The licenses are checked out again when `SIGCONT` is received.


© 2023 Cadence Design Systems, Inc.

You can also set the variable lsusp to select or deselect this check box.

- *License Queue Time Out*—Turns on the queuing for license capability. Spectre sleeps and requests the license again if there is no available license. You have to specify for how long it must wait for a license (in seconds). Specifying value `0` means wait until the license is available.
- *Set Top Circuit as Subcircuit*—Instructs Spectre to netlist the top-level schematic as a subcircuit.
- *Enables Diagnostic Mode*—Enables diagnostic mode that provides debugging information to help you resolve simulation issues related to performance or convergence.

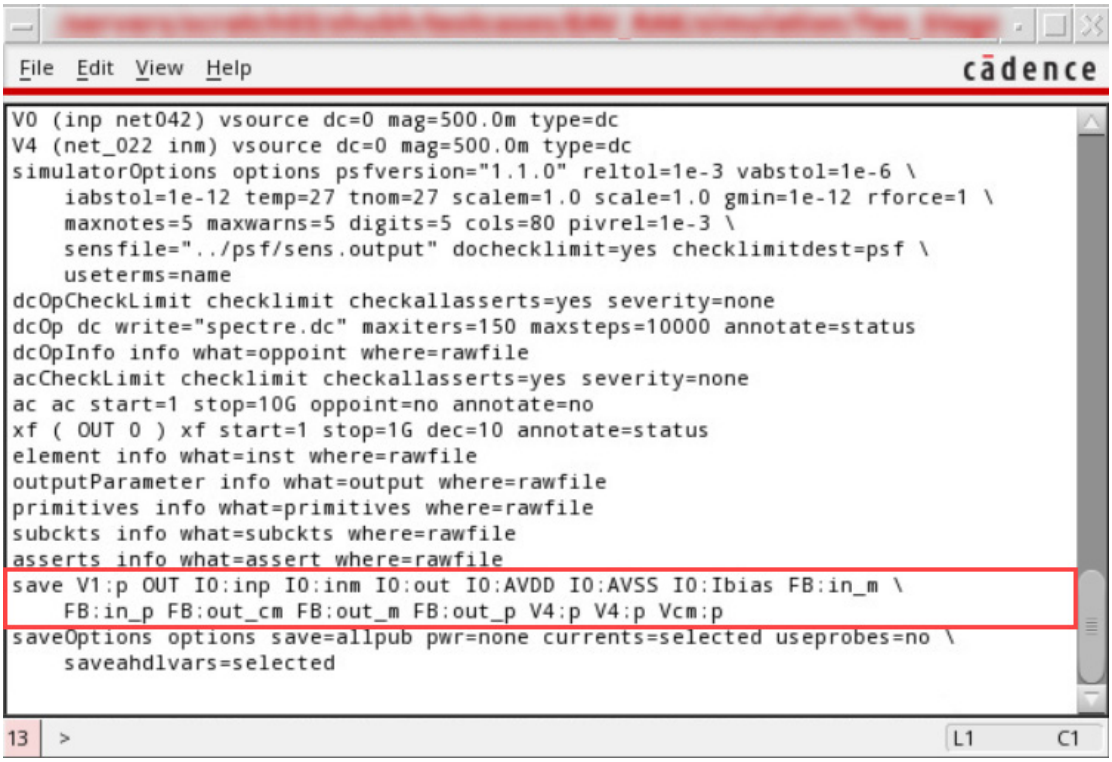
  **Important Points to Note:**

  - This form option will have an impact only when you select sst2 as the output format.
  - If *Using colon as Term Delimiter* option is selected/deselected, you need to netlist the design.
  - *Set Top Circuit as Subcircuit* instructs spectre to netlist the top-level schematic as a subcircuit.
- *PinCheck Term Mismatch Action* and *PinCheck Term Direction Mismatch*—These options are available for spectre netlisting. They are equivalent to the OSS global settings, namely, `simCheckTermMismatchAction` and `simCheckTermDirectionMismatch`. The values set in these fields override the global settings for each test. If these values are not set, then global values take precedence. The valid values are `default`, `ignore`, `warning`, and `error`.

  **Note:** Set it to default to use the properties, `simCheckTermMismatchAction` and `simCheckTermDirectionMismatch`, specified in `.cdsinit` or `.simrc` file. If these properties are not set, then OSS default value, `error`, is used.


© 2023 Cadence Design Systems, Inc.

- *Preserve Subckt Terminal by Names*—Generates the save statements with the instance ports in name format instead of the default index format




For information about how to set the environment options for the AMS simulator, see Chapter 15, “Working with AMS Simulator”.


© 2023 Cadence Design Systems, Inc.

# Setting up Design Variables

You can use design variables and CDF parameters to set component values.

Design variable values are always global to the design. The scope of CDF parameter values, however, depends on which Analog Expression Language (AEL) functions you use to refer to the parameter.

**Note:** Do not use simulator reserved words as design variable names. For more information, see Reserved Words on page 243.

If you use the reserved words for design variables, the following message is displayed:

**Note:** For AMS UNL, all design variables must be defined prior to the netlisting step.

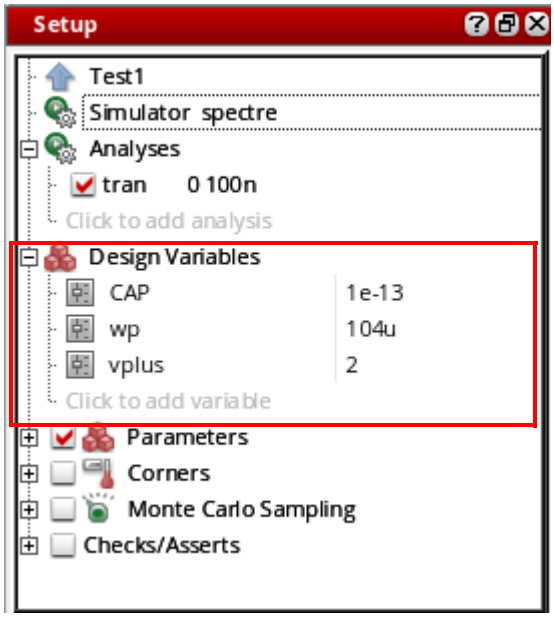
## Adding a New Variable

To add a new variable, do one of the following:

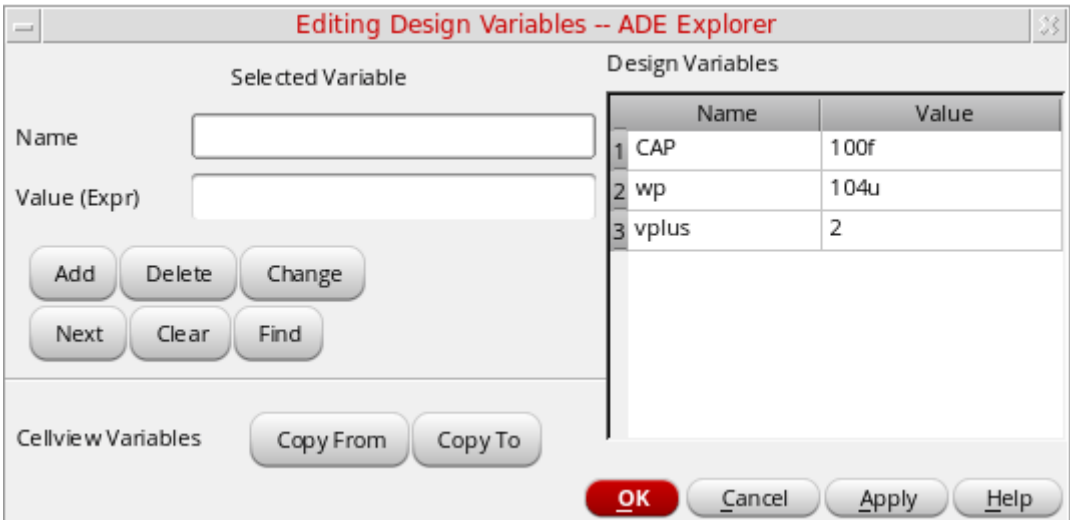
- In the ADE Explorer window, choose *Variables – Edit*.
- On the Run toolbar, click the Edit Variable ( ) button.


© 2023 Cadence Design Systems, Inc.

- In the Setup assistant, in the *Design Variables* section, click the *Click to add variable* text.




The *Editing Design Variables -- ADE Explorer* form appears.




In this form, perform the following steps:

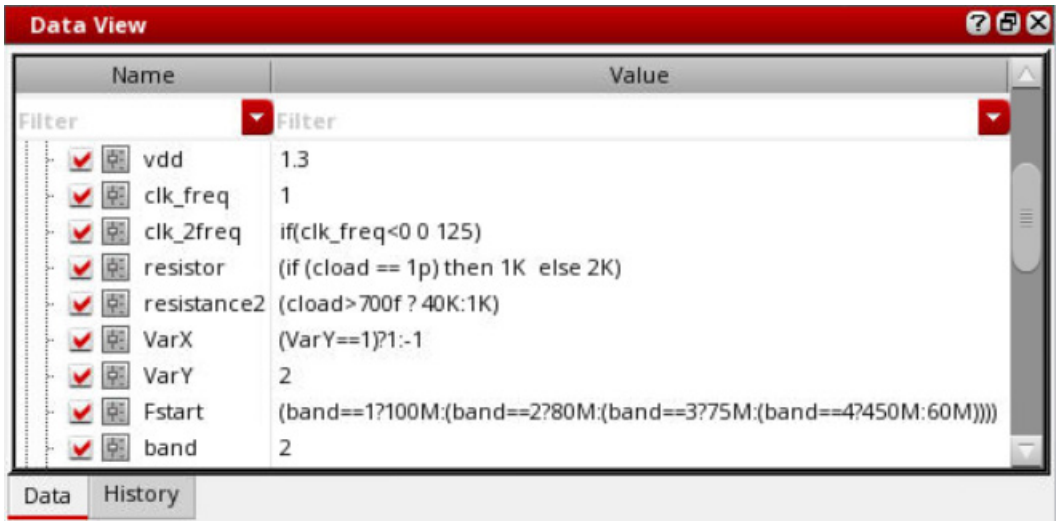
1. Enter the variable name in the *Name* field.

   The name must begin with a letter and can contain only letters and numbers. If the *Name* field already contains the name of a variable, click *Clear*.

2. Enter a number or an expression in the *Value (Expr)* field.


© 2023 Cadence Design Systems, Inc.

A design variable value can also be a conditional expression containing a ternary operator `(?:)`. For example, let us consider the following setup.




- The expression `clk2_freq=if(clk_freq<0 0 125)` denotes that if the value of the variable `clk_freq` is less than `0`, the value of the variable `clk2_freq` is `0`, else it is `125`. Since in this case, the value of `clk_frq` is `1`, the value of `clk2_freq` is `125`.
- The expression `resistance2=(cload>700f ? 40K:1K)` denotes that if the value of the variable `cload` is greater than `700f`, the value of the variable `resistance2` is `40K`, else it is `1K`. Since, in this case, the value of `cload` is `500f`, the value of `resistance2` is `1K`.
- The expression `VarX=(VarY==1)?1:-1` denotes that if the value of the variable `VarY` is `1`, the value of the variable `VarX` is 1, else it is `-1`. Since in this case, the value of `VarY` is `2`, the value of `VarX` is `-1`.
- The expression `Fstart=(band==1?100M:(band==2?80M:(band==3?75M:(band==4?450M:60M))))` denotes that if the value of `band` is:
  - `1`—the value of `Fstart` is `100M`.
  - `2`—the value of `Fstart` is `80M`.
  - `3`—the value pf `Fstart` is `75M`.
  - `4`—the value of `Fstart` `450M`, else it is `60M`.

  Since in this case, the value of `band` is `2`, the value of `Fstart` is `80M`.

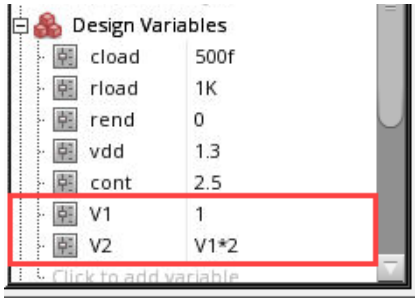

© 2023 Cadence Design Systems, Inc.

The expression can be an equation, a function, or another variable. Expression syntax follows AEL syntax. These expressions are evaluated by the simulator. If the value does not have any number before the decimal point, the AMS Designer simulator inserts a leading zero before the decimal point.

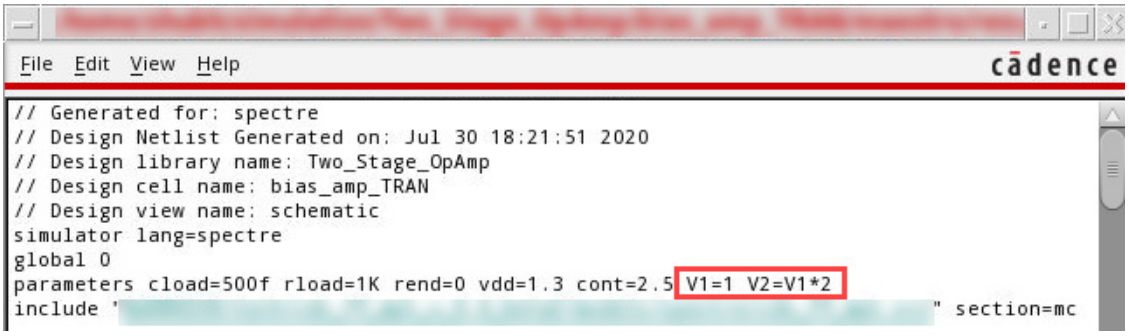
You can also add dependent variables. These are variables based on variables.

For example,

Set the values for two variables, `V1` and `V2`, as `1` and `V1*2` respectively.




These variables are directly written to the netlist, to avoid losing the variable dependencies.



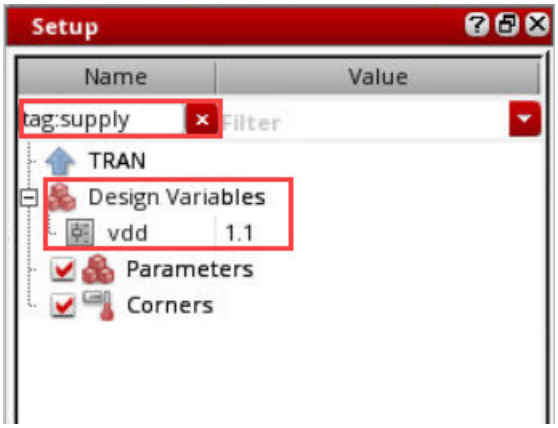

**Note:** If you try to add a new analysis or run a simulation when there is a cyclic dependency between the design variables in the setup, an error message is displayed asking you to edit the design variables to remove that dependency and rerun the simulation.

In the GUI, you can also add a variable without specifying any value for it. The value can be assigned to the variable in a file, which is loaded using a model file at the time of simulation. As the value of the variable is not specified in the GUI, the simulator flags a warning message. For more information, see CheckForNewCellviewVarsUponRun and ignoreUnsetDesVars.

**Note:** On switching from ADE Assembler to ADE Explorer, you can hover above a variable to see the tag assigned to it in ADE Assembler in the tooltip. You can then use these tags to


© 2023 Cadence Design Systems, Inc.

filter the variables using the format `tag: <tagname>`. For example, using the command `tag:supply` in the filter of the *Name* column will filter out all the variables that have the tag `supply` assigned to them.




The following buttons are provided on this form:

- *Add*—Creates the variable you have specified in the *Selected Variable* area.
- *Delete*—Removes a highlighted variable. Click in the list box to highlight a variable.
- *Change*—Updates the highlighted variable with the new information from the *Selected Variable* area.
- *Next*—Highlights the following signal or expression in the *Table of Design Variables* list box.
- *Clear*—Empties the *Selected Variable* area so you can enter a new variable.
- *Find*—Locates the highlighted variable in your design.
- *Cellview Variables*—Keeps variables consistent in the simulation environment and the cellview design database by copying them back and forth.
  - *Copy From*—Copies the variable values in the schematic cellview into the simulation environment.
  - *Copy To*—Copies the variable values in the simulation environment to the schematic cellview.

Additionally, you can also define variables in the definitions file.

- When you click *Add*, the new variable appears in the *Design Variables* table on the right side of the form. You can drag and drop the design variables up/down within the Design Variables section.


© 2023 Cadence Design Systems, Inc.

- If a design variable contains sweep values, such as `vplus=0:2:6`, the tooltip over this design variable in the Setup assistant displays nominal as well as sweep values. However, if you open the Edit Design Variables form, the value for this design variable in the Edit Variable form displays only the nominal value.

  **Note:** If you add a design variable in the schematic and then run the simulation, the design variable automatically gets copied to the ADE Explorer Setup.

### Exporting Variables to a CSV File

To export the design variables from the *Design Variables* section of the Setup assistant to a CSV file:

1. Right-click *Design Variables* and choose *Export to CSV*.

   The Export Variables form appears.

2. In this form, browse to a directory in which you want to save the variables in a CSV file.
3. Specify the name of the CSV file in the *File Name* field.
4. Click *Save* to close the form and export the variables.

All design variables, their values, and enabled or disabled status currently defined in the active setup are saved in the specified CSV file in a standard format, as shown in the example below.

```
Type,Name,Value,Enabled
variable,CAP,50,enabled
variable,rload,10,enabled
variable,cload,20,enabled
```

### Importing Variables from a CSV File

If you already saved variables or design variables in a CSV file, you can import them in the active setup. To import the variables from a CSV file, do the following:

1. Right-click *Design Variables* and choose *Import from CSV*.

   The Import Variables form appears.

2. In this form, browse to a directory from where you want to import the variables saved in a CSV file.
3. Specify the name of the CSV file in the *File Name* field.


© 2023 Cadence Design Systems, Inc.

4. Click *Open* to import the variables from the selected CSV file.

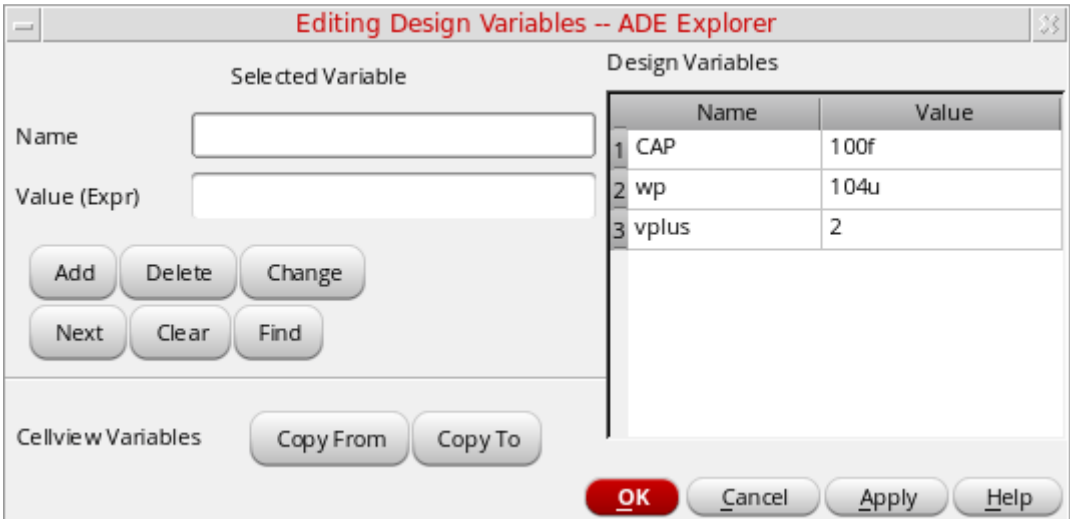
Note the following points:

- If a variable already exists with the same name as that of a variable being imported, it is overwritten by the imported variable.
- The enabled or disabled status of the imported variable is also imported, and set accordingly.
- If you save the global variables in a CSV file using ADE Assembler and then import this CSV file in ADE Explorer, the global variables are imported as design variables and listed in the *Design Variables* section of ADE Explorer. For more information, see Importing Global Variables from a CSV File and Exporting Global Variables to a CSV File in *Virtuoso ADE Assembler User Guide*.
- If the CSV file includes any parametric sets, they are ignored and variables are imported as individual design variables in ADE Explorer.

## Editing Values

To edit the value of a design variable,

1. Double click a variable in the Setup assistant and edit the value in the *Value* field.
2. In the Editing Design Variables form, click the variable name in the Design Variables table.
3. Right-click the variable in the Setup assistant and choose *Edit Variable*.




4. Click the variable name in the *Design Variables* table.


© 2023 Cadence Design Systems, Inc.

The value or expression appears in the *Value (Expr)* field.

5. Edit the value or expression.
6. Click *Change*.
7. Click *Apply* or *Apply & Run Simulation*.

## Deleting Values

To delete a design variable,

1. In the Editing Design Variables form, click the variable name in the Design Variables table.
2. Click the delete button on the Run toolbar.
3. Select the variable to delete in the *Design Variables* section of Setup assistant and choose *Variables – Delete*.

   To select more than one variable, hold down the `Control` key while you click the variables, or click and drag the cursor.

   To de-select a highlighted variable, hold down the `Control` key while you click the variable.
4. Right-click the variable to delete in the Setup assistant and choose *Delete Variable*.
5. Select a variable in the Setup assistant and choose *Variables – Delete*.

## Copying Values From and To Schematic

If you change variables in the Schematic window and want to use these values in your next simulation,

1. Select the design variable in the Setup assistant and choose *Variables – Edit – Copy From Cellview*.
2. Right-click a design variable and choose *Copy From Cellview*.
3. Click the design variable in the Design Variables table of the Editing Design Variables form and click the *Copy From Cellview* button.

If you change variables in the simulation window and want to copy the values back to the cellview before you save the schematic,


© 2023 Cadence Design Systems, Inc.

1. Select the design variable in the Setup assistant and choose *Variables – Edit – Copy To Cellview*.
2. Right-click a design variable and choose *Copy To Cellview*.
3. Click the design variable in the Design Variables table of the Editing Design Variables form and click the *Copy To CellView* button.

**Note:** Make sure that the schematic and simulation environment variables are consistent with each other.

## Displaying Values on the Schematic

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

1. Edit the component's CDF with the CDF Editor.
2. Set *paramEvaluate* to *full*.

## 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:

- In the *Design Variables* section, right-click a design variable that you want to locate in the schematic 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.

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 is displayed in CIW.

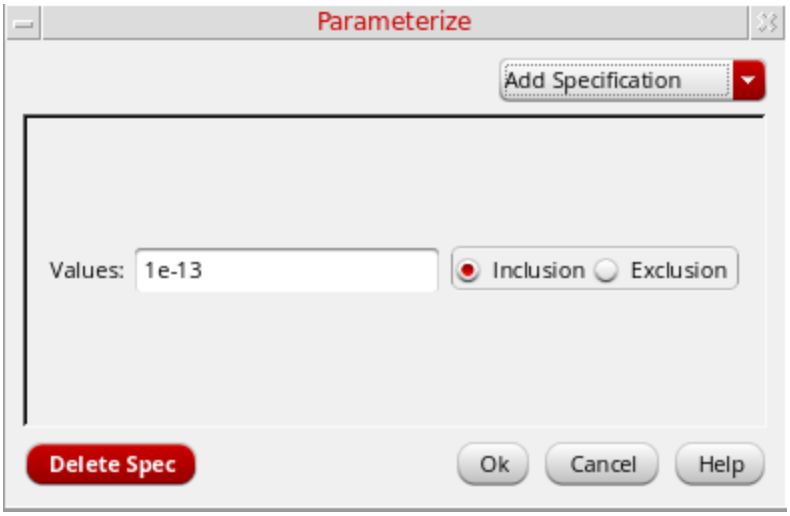
**Note:** You can drag the design variables to change their order in the Setup assistant.

## Performing Parameterization in Design Variables

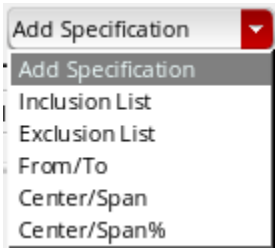
You can set up a sweep directly in the value field of the design variable in the `start:step:stop` format. For example, var1= `2:1:10`, where start value=`2`, end value=`10`, and step size=`1`.


© 2023 Cadence Design Systems, Inc.

To parameterize a design variable, double-click the value of the design variable in the Setup assistant. The ellipses button appears. When you click this button, the Parameterize form appears.




In this form the drop-down at the top consists of the following options:




- *Inclusion List*: Specifies a list of specific points in addition to the range you have specified. You can specify any number of additional points.
- *Exclusion List*: Specifies a list of specific points to be excluded from the range you have specified. You can specify any number of points. If a value is specified in both inclusion and exclusion list, the value is excluded from the sweep sets.
- *From/To*: Lets you specify the limits of the sweep range with numerical values. If you do not specify any value in the *From* column, the default value `0` is taken. If you do not specify any value in the *To* column, only the start value (specified in the *From* column) is considered and steps are not calculated. If you do not specify any value in both the *From* and *To* columns, simulation is not done and an error is flagged.
- *Center/Span*: Lets you specify a center point and the range of values around the center you want to sweep. For example,
  *center* = 100, *span* = 20 is equivalent to *from* = 90, *to* = 110.


© 2023 Cadence Design Systems, Inc.

- *Center/Span%*: Lets you specify a center point and a range around the center. With this option, you can specify range limits as a percentage of the center value. For example, *center* = 100, *span%* = 40 is equivalent to *from* = 80, *to* = 120.

  If you do not specify any value in the *Center* column, the default value `0` is taken. If you do not specify any value in the *Span* or *Span%* column, only the start value (specified in the *Center* column) is considered and steps are not calculated. If you do not specify any value in both the *Center* and *Span* or *Span%* columns, simulation is not done and an error is flagged.

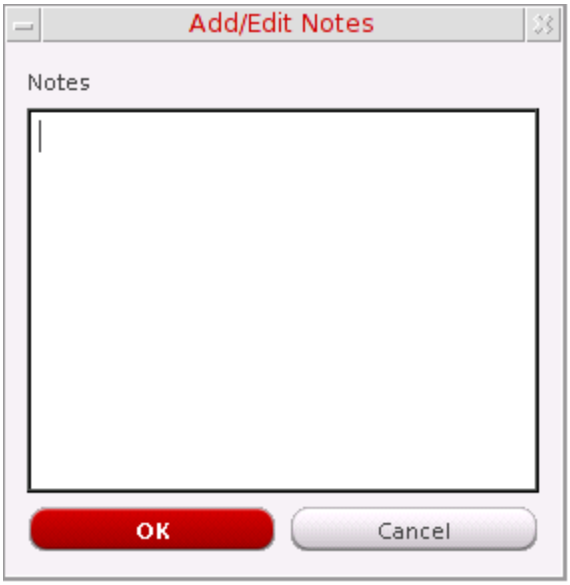

© 2023 Cadence Design Systems, Inc.

# Adding Notes

To add notes to a test, do the following:

1. In the Setup assistant, right-click the test name and choose *Notes*.

   The Add/Edit Notes form appears.




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 Explorer setup database. Similarly, you can add notes for parameters and corners.

### *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 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 the test name and choose *Delete Notes*.


© 2023 Cadence Design Systems, Inc.

# Creating and Displaying Netlists

The ADE Explorer creates or updates the simulator input file automatically when you give the command to run a simulation.

You do not need to use the *Netlist – Create* command unless

- You want to use ADE Explorer to create a netlist but run the simulator in standalone mode
- You want to modify the netlist, perhaps to take advantage of features that the design environment interface to your simulator does not support
- You want to read the netlist before starting the simulation

There are two kinds of files generated for simulation:

- The netlist, which contains component information but no simulation control data
- The simulator input file, which contains both the netlist and the simulator control information required (this file is passed to your simulator)

Netlists are hierarchical. They are created incrementally, re-netlist only the changed schematics in a design. All schematics can be forced to re-netlist by using the following commands:

- Choose *Simulation – Netlist – Recreate*.
- Right-click the test name and choose *Netlist – Create or Recreate*.

**Note:** You can run multi-point simulations in ADE Explorer that generate multiple netlists. However, when you choose the *Netlist – Create/Recreate* command, it shows a single point netlist that is created in a temporary directory. While running a simulation, this temporary directory is used as an input to perform Incremental Netlisting.

## The .simrc File

You can use the `.simrc` file with Spectre during interface initialization to customize netlisting. This file helps you set or override defaults for simulation variables in such a way that the changes affect only your own simulations. The `.simrc` file is a way to set defaults on a per-user or per-system basis, and no other designer is affected by this file. This file is optional and is loaded if it exists. It is searched for in the following order:

```
$SIMRC/.simrc
$ossSimUserSiDir/.simrc
dfII/local/.simrc
Current UNIX directory/.simrc
~/.simrc
```


© 2023 Cadence Design Systems, Inc.

The search stops at the first place where the file is found. No other `.simrc` files are loaded unless the load is done from within the first one it finds, allowing for tiered loading or for the local CAD group to alter or disallow the search mechanism. If it does not exist, no error is generated. You can delete the `.simrc` file if you do not want it to be taken into account while netlisting.

## Incremental Netlisting

Incremental netlisting is faster than full hierarchical netlisting because only the schematics that have changed since the previous netlist was generated are re-netlisted. This substantially speeds up netlisting of hierarchical designs containing many small schematics. The system keeps track of the status of each schematic during and between design sessions.

## Creating and Displaying a Netlist

To create and display a new netlist,

- Choose *Simulation – Netlist – Create*.
- Right-click the test name and choose *Netlist – Create*.

  The simulator input file is created in a file. The name of the file is `input` with the simulator-specific extension. For Spectre, the extension is `.scs`. This file is put in the following directory:

  *projectDirectory*`/`*topCellName*`/`*simulatorName*`/`*view*`/netlist/input.`*ext*

  - *projectDirectory* is the directory you specified in the Choosing Simulator form.
  - *topCellName* is the root level cell name of the design.
  - *simulatorName* is the name of the simulator.
  - *view* is the view name being netlisted.
  - *ext* is the simulator-specific extension.

After successful creation of the netlist file, it is displayed in a new window.

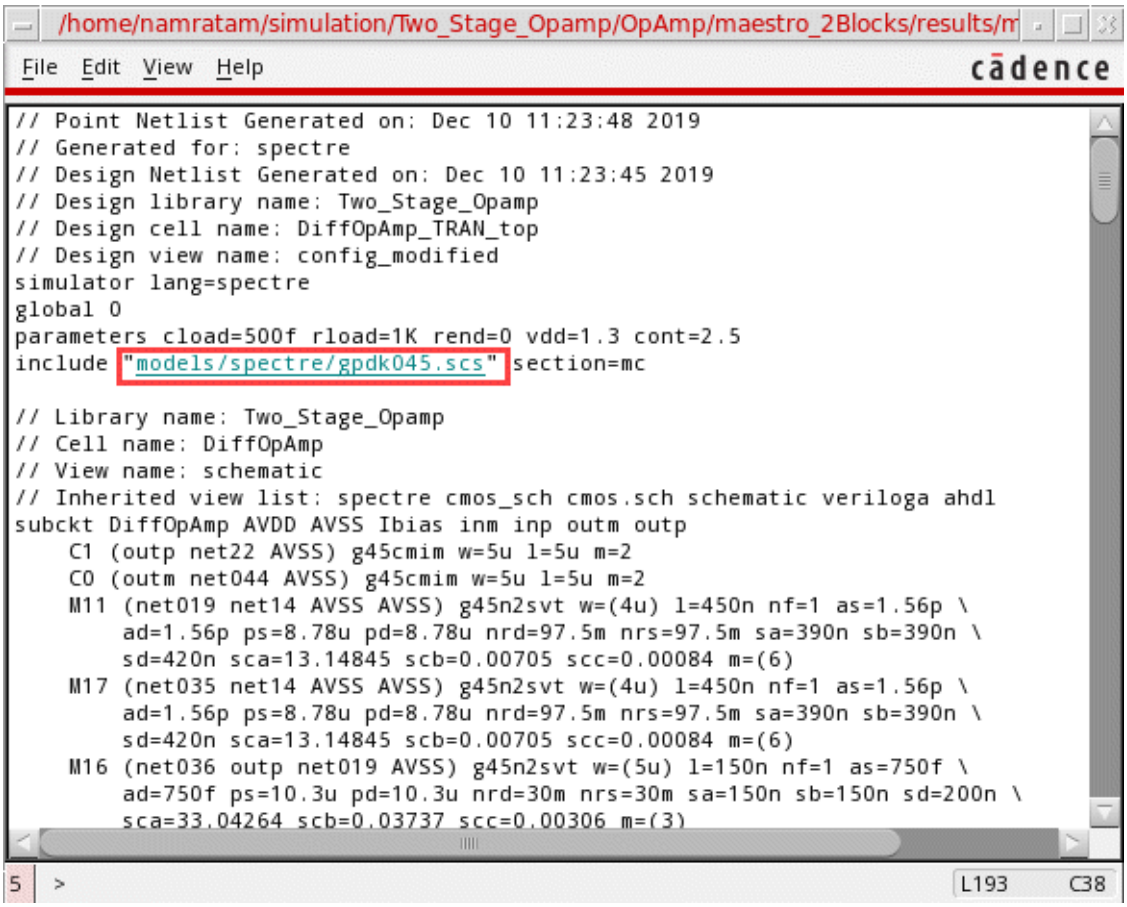
To display an existing netlist file, do any of the following:

- Choose *Simulation – Netlist – Display*
- Right-click the test name and choose *Netlist – Display.*


© 2023 Cadence Design Systems, Inc.

The netlist is displayed in a new window. Review the design components, their connections, and the files to be included in the simulation run. Note that the netlist contains hyperlinks to `.scs`, `.va`, `.mod`, `.lib`, `.dspf`, `.fl`, `.vcd`, `.evcd`, `.mod`, `.def`, `.info`, `.log`, `.vec`, `.v`, `netlist`, `pspice`, `SPEF`, and `DSPF` files.

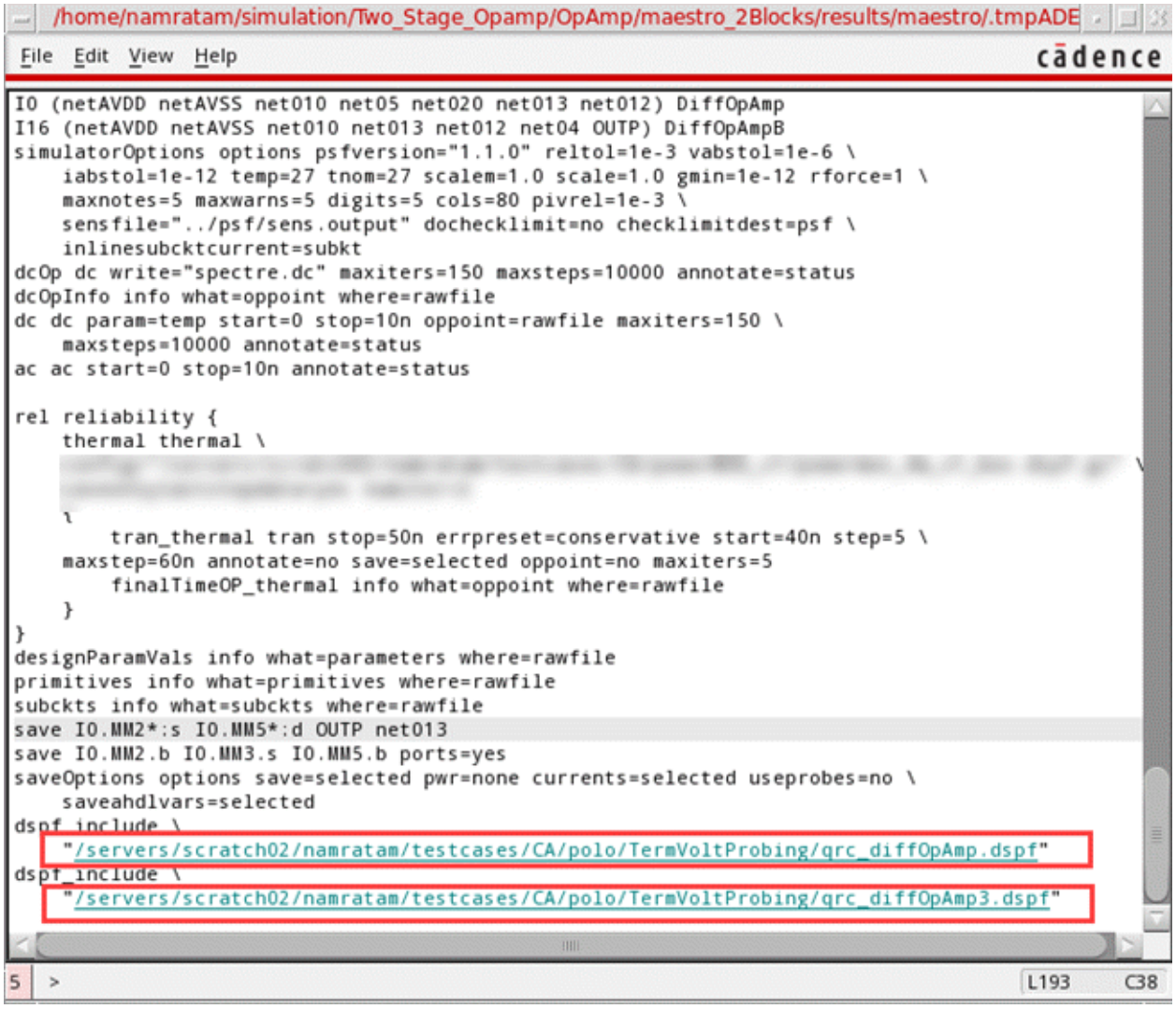
For example, if you have used a model file, you can see a hyperlink pointing to it as highlighted in the following image.





© 2023 Cadence Design Systems, Inc.

Similarly, if you have used a `.dspf` file, the following image shows a related hyperlink.




ADE Explorer creates multiple netlists for a multi-point simulation run. However, when you choose *Netlist – Create* or *Recreate*, only a single-point netlist created in a temporary directory is displayed.

If you want to validate the successful creation of the netlist, run the *Netlist – Create* command instead of running a simulation.


© 2023 Cadence Design Systems, Inc.

# Variables for Customizing Netlist Generation

You can customize ADE Explorer netlisting by setting the following variables.

| Variable | Description |
|---|---|
| `nlReNetlistAll` | When set to t, all cellviews in the design are re-netlisted, irrespective of the setting specified using the ADE GUI options *Create* or *Recreate* in the *Simulation – Netlist* submenu. If `nlReNetlistAll` is not set and has its default value, `nil`, the ADE GUI netlisting options are used. You can set this variable in the `.simrc` file. |
| `globalGroundNet` | When you use this variable to specify a string, the value specified is considered to be the global ground net and overrides the default analogLib global ground net (gnd!). You can set this variable in `.simrc` file. This variable is valid for socket direct netlister only. |
| `hideGlobalGroundNetMessage` | When set to `t`, this variable suppresses the information messages issued after you change the default value of the `globalGroundNet` variable.<br><br>The default value of `hideGlobalGroundNetMessage` variable is `nil`. |
| `simPropValueNotation` | When set to `Engineering`, all property values are displayed in engineering notation in the netlist. For Example, a resistance value of `1m` is displayed as `1e-3`. If `simPropValueNotation` is set to `Scientific`, all property values are displayed in scientific notation in the netlist. For Example, a resistance value of `1m` is displayed as `1.000000000e-03`. You can set this variable in si.env file. This variable is valid for socket netlisters only. |
| `hnlDynamicFlagOn` | This variable can be set to `t` or `nil` in the `.cdsinit` file. If the variable is not specified, it defaults to `nil`. If set to `t`, the parameter values having length more than 8K, which is the maximum limit, are printed in the netlist.<br><br>If set to `nil`, the parameter values with length more than 8K are truncated to the maximum limit. |


© 2023 Cadence Design Systems, Inc.

`spfHnlMapNetInName`

Use this variable in the `.simrc` file to specify mapping rules for bus nets in the net name.

For example, you can escape `$, ;` and `:` with `\` in the net names.

```
spfHnlMapNetInName =  '( "$" ";" ":"  )
```

With this setting, net name `I0.netA$` is printed in the netlist as `I0.netA\$`.

In addition, you can also specify escaping characters if you do not want to use `\`, as shown below.

```
spfHnlMapNetInName =  '( "$" "," ";" ":"
"=" "/" " " "\t" (">" "\]") ("<" "\[")  )
```

Here, `<` and `>` in net names are mapped to `\[` and `\]`. For example, net name `I0.net10<0>` is mapped to `I0.net10\[0\]`.

`spfHnlMapInstInName`

Use this variable in the `.simrc` file to specify mapping rules for iterated instances in the instance name.

For example, you can escape a space and `/` in the instance names with `\`.

```
spfHnlMapInstInName =  '( " " "/")
```

With this setting, instance name `Y/AN_n` is printed in the netlist as `Y\/AN_n`.

In addition, you can also specify escaping characters if you do not want to use `\` , as shown below.

```
spfHnlMapInstInName =  '(" " "/" (">" "\]")
("<" "\[")  )
```

For example, instance name `Y_n<0>` is mapped to `Y_n\[0\]`.


© 2023 Cadence Design Systems, Inc.

`nlCustomSubcktModelNameSuffix`

Use this variable in the .simrc file to add a custom suffix to the model name generated for the subcircuits belonging to the specified library and cell. This avoids conflict with the subcircuit model name in case there is an external spice file bound to the same view.

For example, on using the following setting, the ADE netlister prints the subcircuit model name for `DiffOpAmp` as `DiffOpAmp_NoSpiceSuffix`, avoiding a clash with the subcircuit model name for `DiffOpaAmp` in the external spice file included in the HED binding.

```
nlCustomSubcktModelNameSuffix =
list(list("Two_Stage_Opamp" "DiffOpAmp"
"NoSpiceSuffix"))
```

`nlSchPcellPerCellCustomFormat`

Use this variable in the .simrc file to specify the rules for customizing the subckt names for multiple Pcells at once.

For example, let us specify the following setting in the.simrc file.

```
nlSchPcellPerCellCustomFormat =
list(list("mylib" "hp_analog_cell"
"schematic" "l:%NAME,sfy_b:%NAME
%VALUE,w:%VALUE,spotr2:%NAME-%VALUE")
list("mylib" "analog_cell" "schematic"
"l:%VALUE,ploda1:%NAME-
%VALUE,w:%NAME,spotr2:%NAME"))
```

The subcircuit names are then updated in the netlist as follows.

```
subckt
hp_analog_cell_pcell_l_sfy_b134_00n_146_00
n_spotr254_5n_11803695165159852977   D G S
B
```

```
subckt
analog_cell_pcell_6n_ploda16n_w_spotr2_281
9258373539267837D G S B
```


© 2023 Cadence Design Systems, Inc.

| | |
|---|---|
| `nlCustomMasterOptionList` | Use this variable in the .simrc file to trigger auto-netlisting when a change is detected in any of the options specified through this variable.<br><br>For example, let us specify the following setting in the .simrc file.<br><br>`nlCustomMasterOptionList = list(list("hnlProcessNetsInReverse" "global") list("nlCustomerSubcktPrefix" "global" ) list("liclogs" "env"))`<br><br>If any of the specified environment variables are updated, the re-netlisting process gets triggered. |
| `nlCustomizedTermOrderList` | The `nlCustomizedTermOrderList` variable is used to customize the terminal order in netlist. The value of this variable is a list that represents the possible combinations for portorder and defines the preference order. The available terminal orders are: CDF `simInfo` termOrder (`cdfTermOrder`), schematic portorder (`masterPortOrder`), name of the custom function for fetching terminal order (`customFunc`), and alphanumeric order given by OSS (`defaultTermOrder`).<br><br>`nlCustomizedTermOrderList = list("cdfTermOrder" "masterPortOrder" "customFunc" "defaultTermOrder")` |


© 2023 Cadence Design Systems, Inc.

# Setting up Stimuli

There are three ways to set up stimuli in the simulator:

- Add source symbols to the schematic
- Use the Stimuli Assignment form
- Specify a stimulus file

## Setting up Stimuli Using the Stimuli Assignment Form

This form is used to define stimuli and assign them to your design. It creates a simulator input file that is used by the simulator.

For input stimuli, your top-level schematic must contain input pins for the signals that you plan to set. To use the power stimuli, you must use a global name on a signal (such as vdd!).

All sources, whether used for stimulus or for a power supply, are assumed to come from the `analogLib` library, a library supplied by Cadence. If your sources are located in a different library, you must add the *refLibs* property to your design library to specify the location of the source information.

It involves three steps, that of, setting up the simulation environment for external stimuli, creating a simulator input file, and then generating a stimulus file containing input and power source stimuli in the proper syntax for your simulator.

1. To access libraries other than the default `analogLib` library for sources:

   **a.** From the CIW, choose *Tools – Library Manager*.

   The Library Manager: Directory form appears.

   **b.** Select the library name of the current design.

   **c.** Choose *Edit – Properties*.

   The Library Property Editor form appears.

   **d.** Verify that the *refLibs* property has been set to the appropriate library search sequence. This property should be visible in the lower section of the form.

   **e.** Click *OK*.

   If there is no *refLibs* entry, click *Add* on the Library Property Editor form, add the data specified below,


© 2023 Cadence Design Systems, Inc.

In the Add Property form, specify the following property name and characteristics.

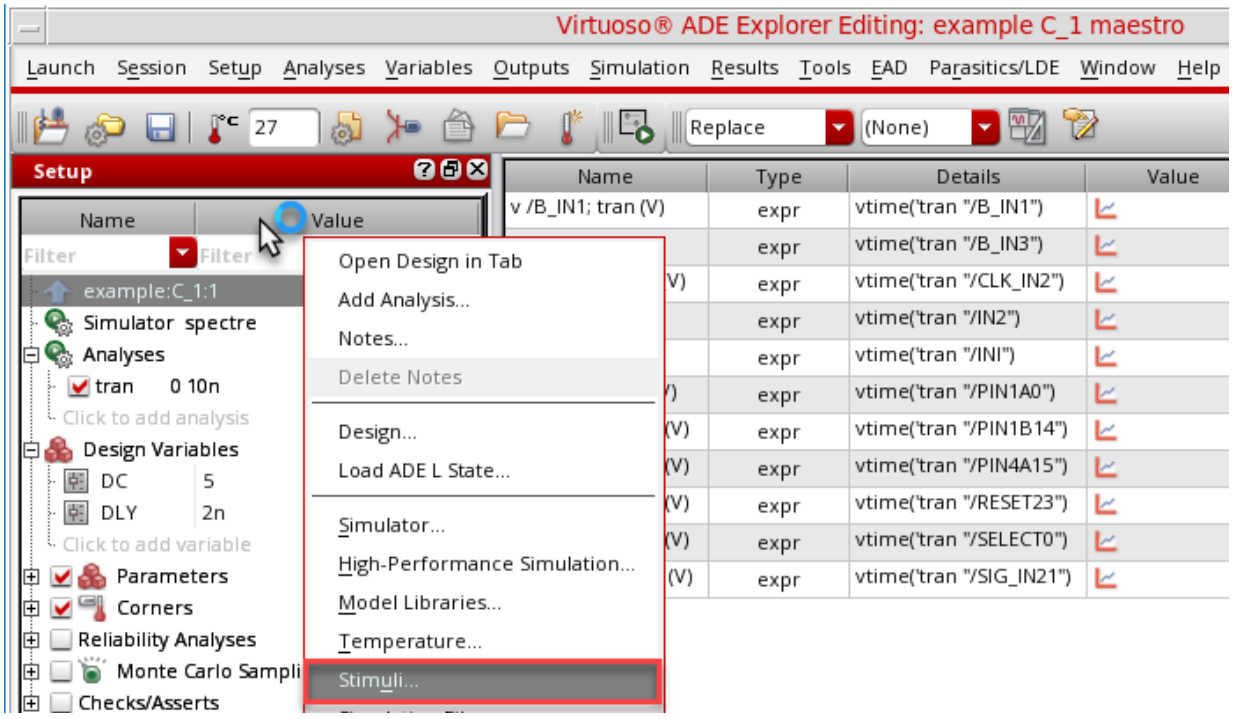
| | |
|---|---|
| Name | *refLibs* |
| Type | *string* |
| Value | list of one or more libraries in the required search sequence from left to right |

The *refLibs* property and the search list are displayed in the parameter list on the Library Property Editor form.

**f.** Click *OK* to return to the Library Manager form.

**2.** Choose *Setup – Stimuli* in the Simulation window to add stimuli.

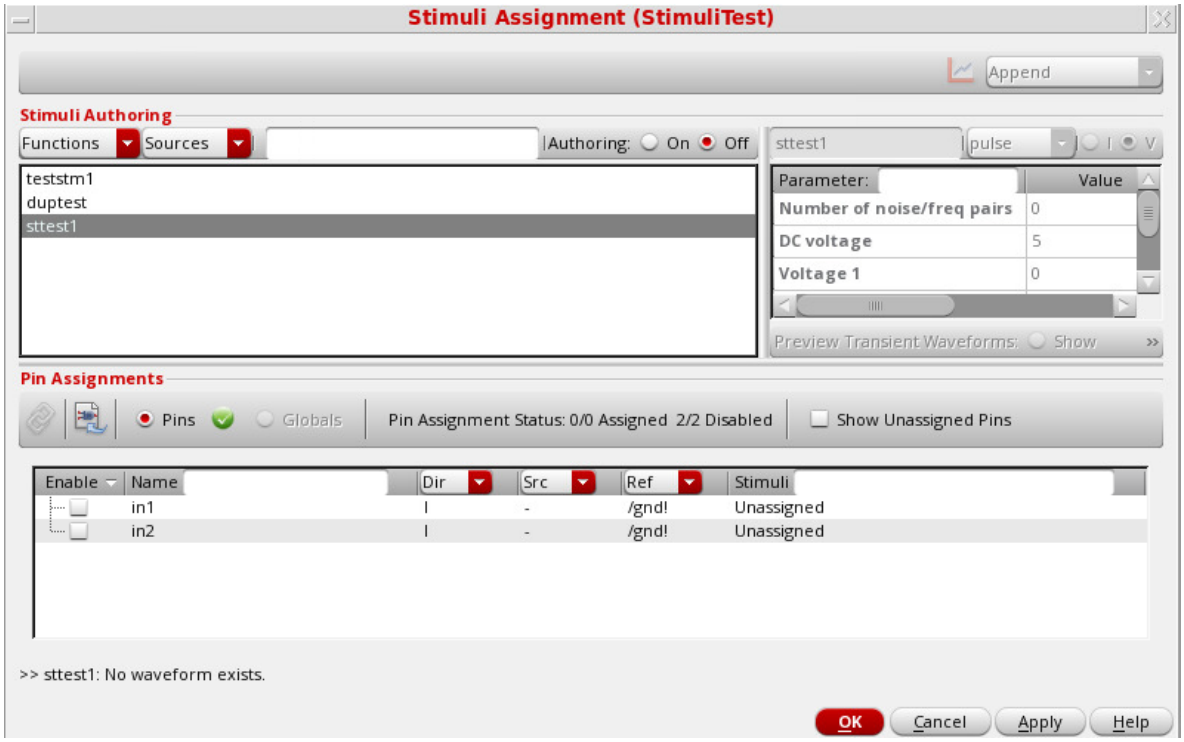
Alternatively, right-click the test name in the Setup Assistant or the Dataview Assistant in ADE Assembler and choose *Stimuli*.





© 2023 Cadence Design Systems, Inc.

The Stimuli Assignment form appears.




The form consists of two sections:

- Stimuli Authoring - Creates stimuli and helps you manage them
- Pin Assignments - Assigns stimuli to pins and global nets

## Creating and Managing Stimuli

You can see the predefined stimuli in this section. These could have been created for the view specified or imported from another design.

### Creating Stimuli

To create new stimuli:

1. In the *Stimuli Authoring* section, select *Authoring* `On`. This enables the authoring section of the form and disables pin assignments.
2. Enter the stimulus name in the text field. The name should be alphanumeric.

   **Note:** It the stimulus name is illegal, the text field is highlighted in red.

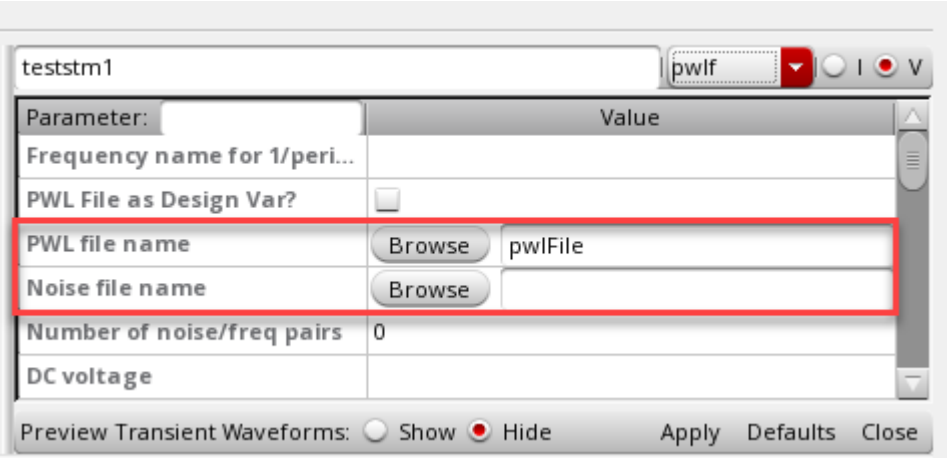

© 2023 Cadence Design Systems, Inc.

3. Select the type of stimulus.

## Type of stimuli

| Type | Description |
|---|---|
| `bit` | Bit |
| `dc` | Direct current |
| `sin` | Sinusoidal waveform |
| `pulse` | Pulse waveform |
| `exp` | Exponential waveform |
| `pwl` | Piecewise linear waveform |
| `pwlf` | Piecewise linear waveform file |

**Note:** You can browse to find the required files for different types of stimuli.




4. Specify whether it is a current or voltage source .
5. Modify the values of stimulus parameters. These values can also be specified as expressions.

   **Note:** If the specified expression has an error, it gets highlighted in red and the *Apply* option gets disabled.
6. Select *Show* in *Preview Transient Waveforms* to enable *Stop Time.*
7. Specify the stop time to generate the waveform.

   **Note:** It can be set using a valid value, design variable or an environment variable


© 2023 Cadence Design Systems, Inc.

stopTime. The default value of this environment variable is 1n.

8. Click *Apply*.

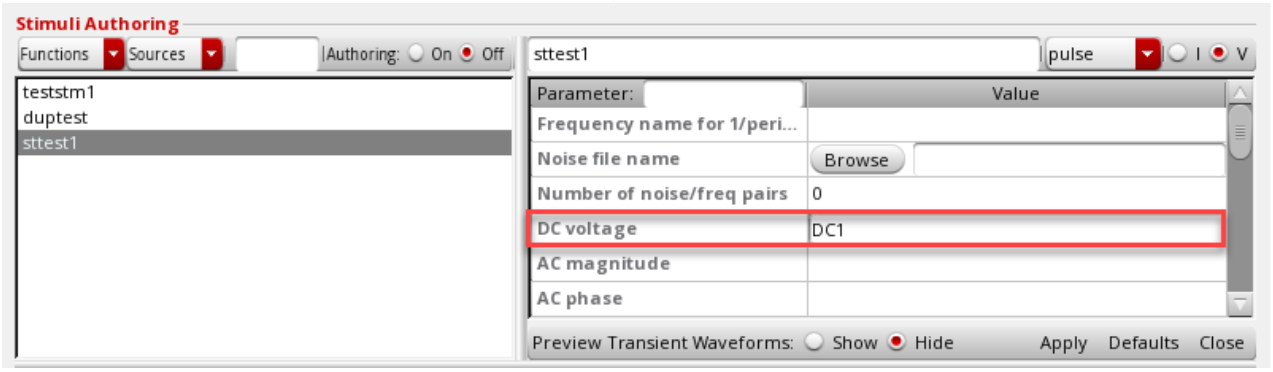
   Notice a new stimulus entry in the top-left section.

**Note:** To edit an existing stimulus, you can search a stimulus by its name. Select the stimulus from the list and make required changes.

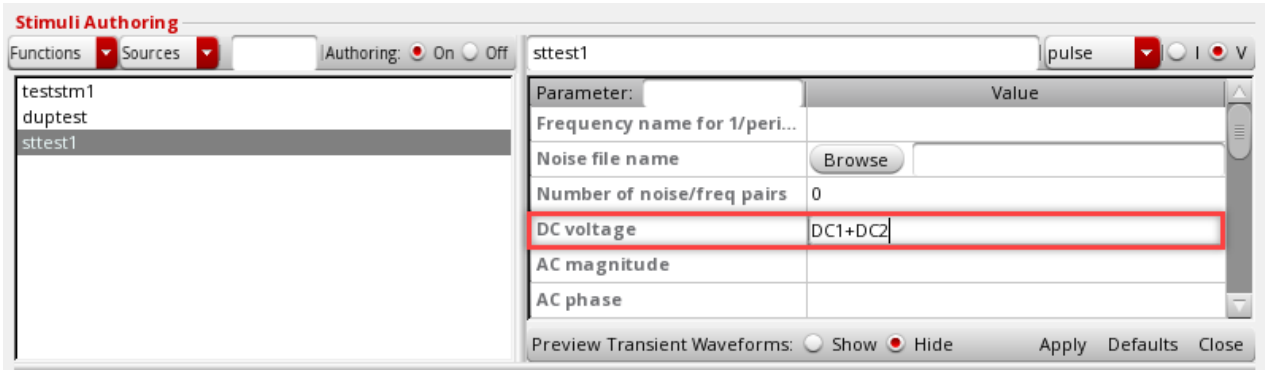
All the stimuli definitions are stored on the path — `<Lib-name>/<Cell-name>/<View-name>/namedStimuli/stimuli.xml`.

### *Using Variables to Set Parameters of Stimuli*

You can use design variable names defined in ADE Explorer to set a parameter value in the stimuli definition. For example, the DC voltage is set to DC1 in the specified stimulus.



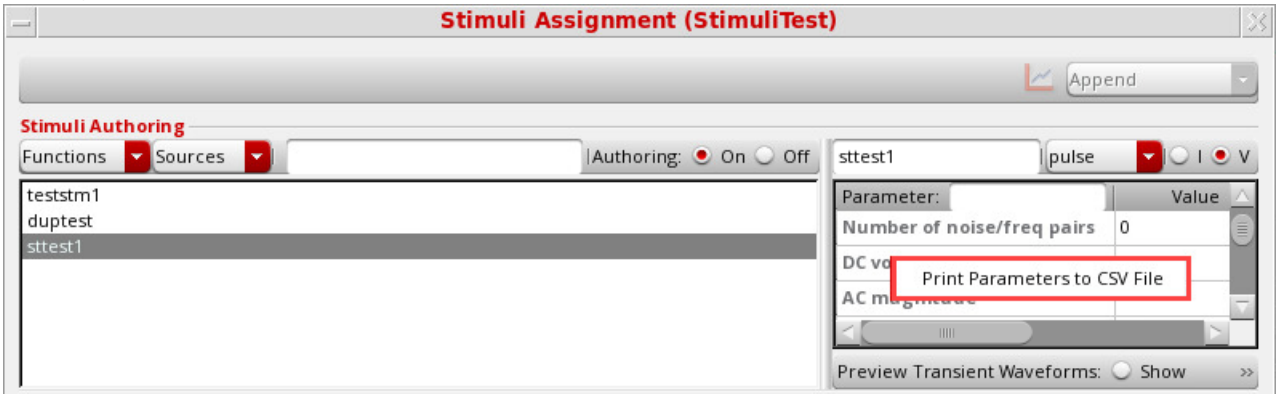

You can also use an expression containing design variables to define a parameter value in stimuli.





© 2023 Cadence Design Systems, Inc.

To print the details of the specified parameters under the *Stimuli Authoring* section to a CSV file, right-click a parameter in the list box and choose *Print Parameters to CSV File*.




The *Save Parameters to CSV File* window is displayed.Either select an existing file or create a new one and specify its name in the *Field_name* field and select *Save*.

The parameters are then saved in that file in CSV format.

*Important*

Any variable added to a stimulus definition in the format `VAR("<myVariable>")` is automatically added to the Design Variables in ADE Explorer or in the Data View Assistant in ADE Assembler. It prompts you to assign values to such variables.

If the environment variable, autoPromoteVarsToGlobal, is set to t, the new variable is added to the list of Global variables.

### *Importing Stimuli*

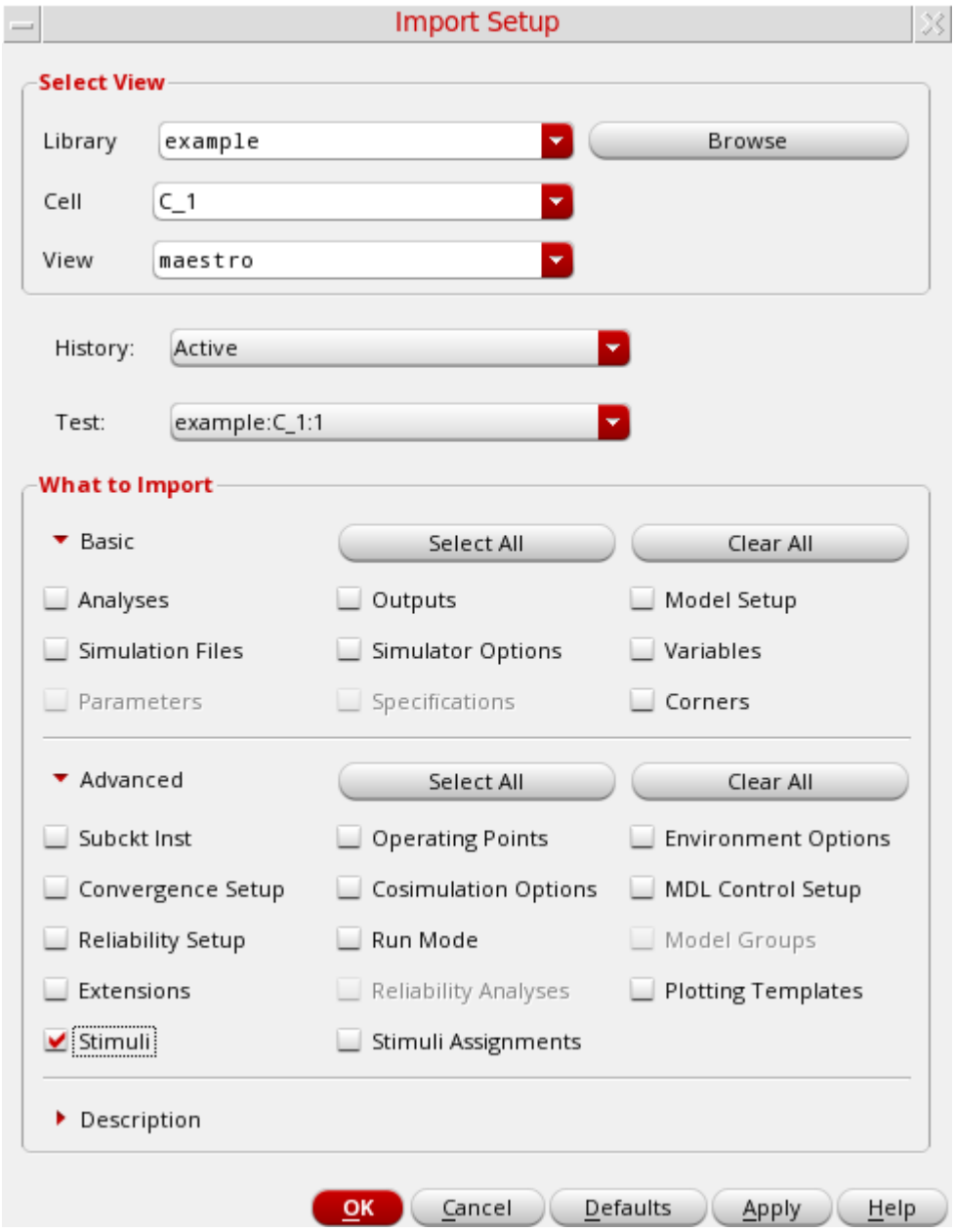
To import stimuli from other cellviews, follow the steps:

1. In ADE Explorer, choose *Session – Import*.
2. Select the Library, Cell, and View to import their stimuli.
3. Select the check box, *Stimuli*.

The imported stimuli definitions are appended to the existing stimuli definitions in your view. For example, you have two stimuli definitions, namely, `pulse1` and `vsin1` in your design, then you import stimuli from another view. You will still have `pulse1` and `vsin1`, and the other stimuli definitions will be added.


© 2023 Cadence Design Systems, Inc.

**Note:** The stimuli assignments can be imported in ADE Explorer. These will overwrite the current assignments.




*Important*

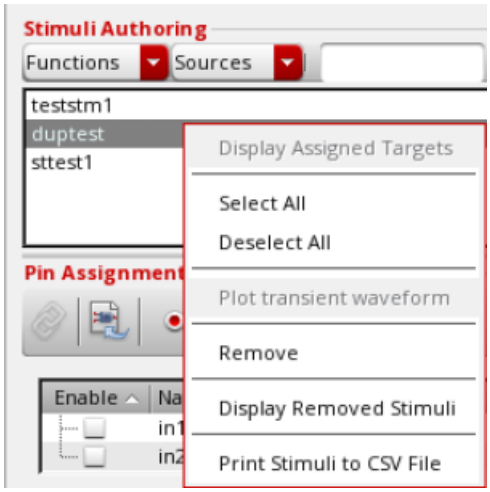
To generate waveforms for all the imported stimuli definitions in one go, run the command `maeStmGenerateWaveforms()` in CIW. This command also generates waveforms for any stimuli definition that is out of date, for example, a stimulus uses a design variable whose value has now changed.


© 2023 Cadence Design Systems, Inc.

### *Removing Stimuli*

Stimuli can be removed from this form. However, there is also a safety net to retrieve the removed stimuli.

To remove a stimulus, right-click the stimulus name and select *Remove*.

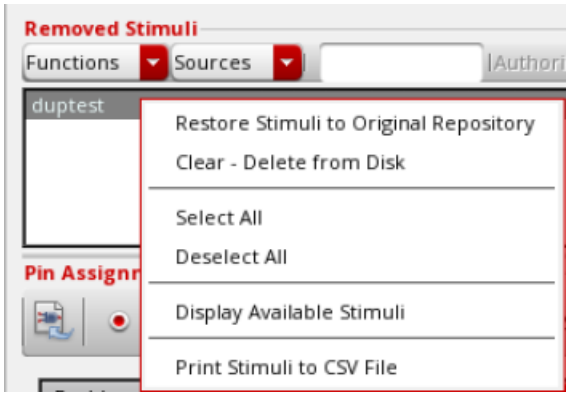



**Note:** If a stimulus is assigned to a pin, the option to remove it from the form is disabled.

To display all the removed stimuli, right-click in the *Stimuli Authoring* section and select *Display Removed Stimuli*.

You can perform the following actions on the removed stimuli:

- Restore the removed stimuli –Right-click the stimulus name and select the *Restore Stimuli to Original Repository* option. This restores the removed stimulus to the original stimuli list. To see the available stimuli, select *Display Available Stimuli*.
- Clear/ Delete the removed stimuli –Right-click the stimulus name and select *Clear - Delete from disk*.





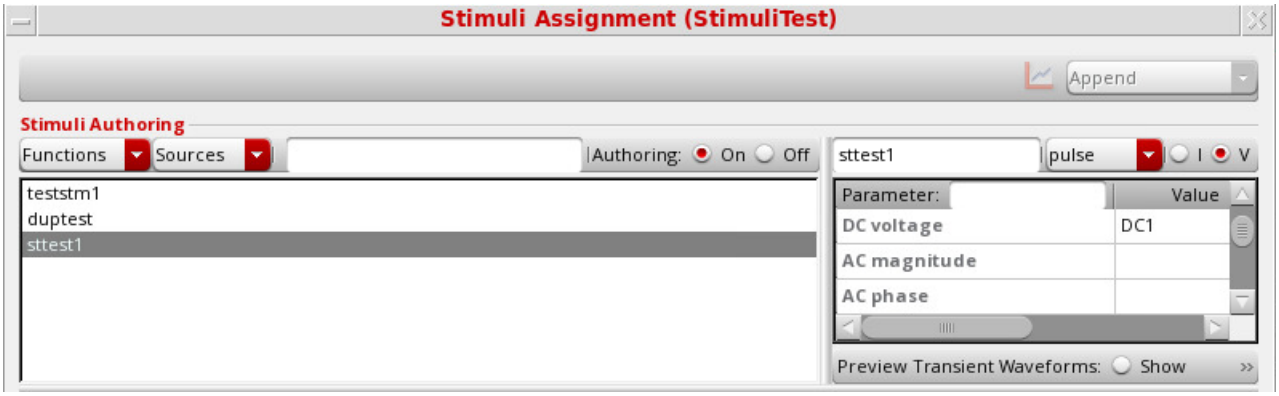
© 2023 Cadence Design Systems, Inc.

The `stimuliRemovedNames.xml` contains information of the removed stimuli that have not yet been deleted from the disk.

The waveform information is stored in the simulation directory. The location of this simulation directory is set by the following cdsenv

```
envSetVal("asimenv.startup" "projectDir" 'string "./simulation")
```

It is only created if the *Apply* button is clicked in the *Preview Transient Waveforms* section in the *Stimuli Assignment* form.




**Note:** you remove stimuli but do not delete them from disk, they remain available for import or export with the associated maestro views.

### Editing Stimuli

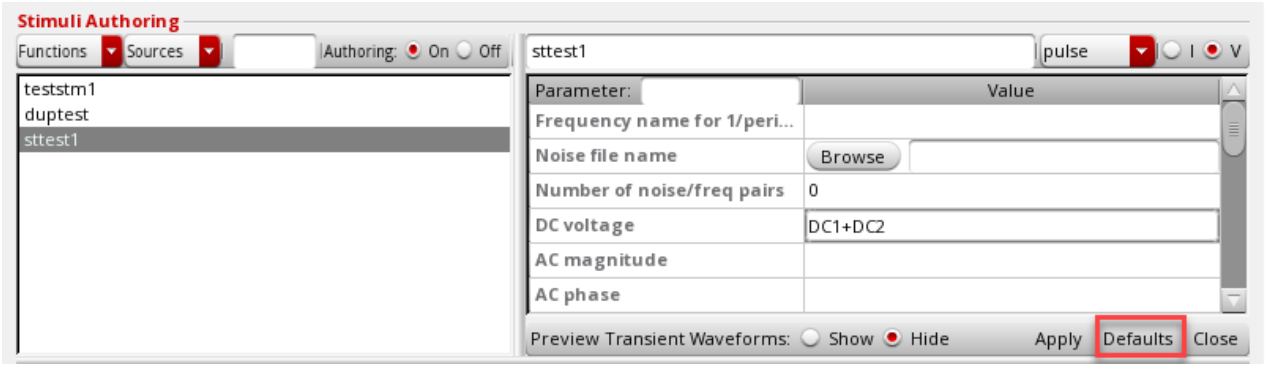
To edit an existing stimuli:

1. In the *Stimuli Authoring* section, select *Authoring* `On`.
2. Modify the required parameter values.
3. Regenerate the waveform. To learn about generating waveform, refer to the section Previewing Stimuli Waveform below.
4. Click *Apply*.


© 2023 Cadence Design Systems, Inc.

*Important*

Click *Default* to reset the stimulus parameters to their default values from the CDF.




### Creating Copy of Existing Stimuli

To create a copy of an existing stimulus, follow these steps:

1. Select the stimulus in *Stimuli Authoring* section.
2. Select *Authoring* `On`.
3. Rename the stimulus.

   You can edit its parameters values and save as a new stimulus.

### Printing the Stimuli details to a CSV file

To print the stimuli details to a CSV file, follow these steps:


© 2023 Cadence Design Systems, Inc.

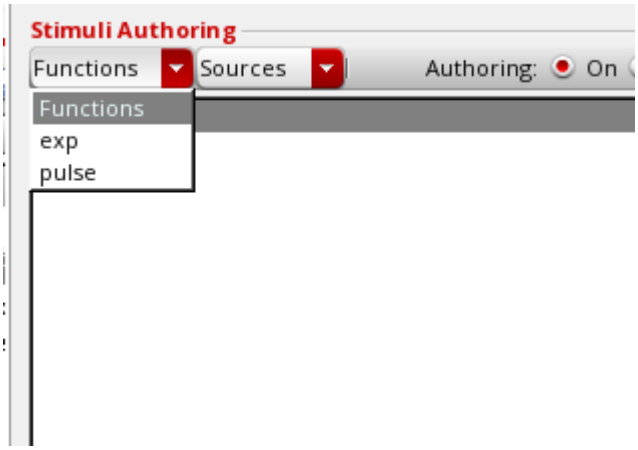
1. Right-click a stimuli entry in the list box under the *Stimuli Authoring* section and choose *Print Stimuli to CSV File.*

2. The *Save Stimuli CSV File* window is displayed. Either select an existing file or create a new one and specify its name in the *File_name* field and click *Save*.

   The contents of the list box will then be saved in that file in CSV format.

*Tip*

To edit stimuli, you can filter them according to their type or source to identify the most relevant set of stimuli for you.



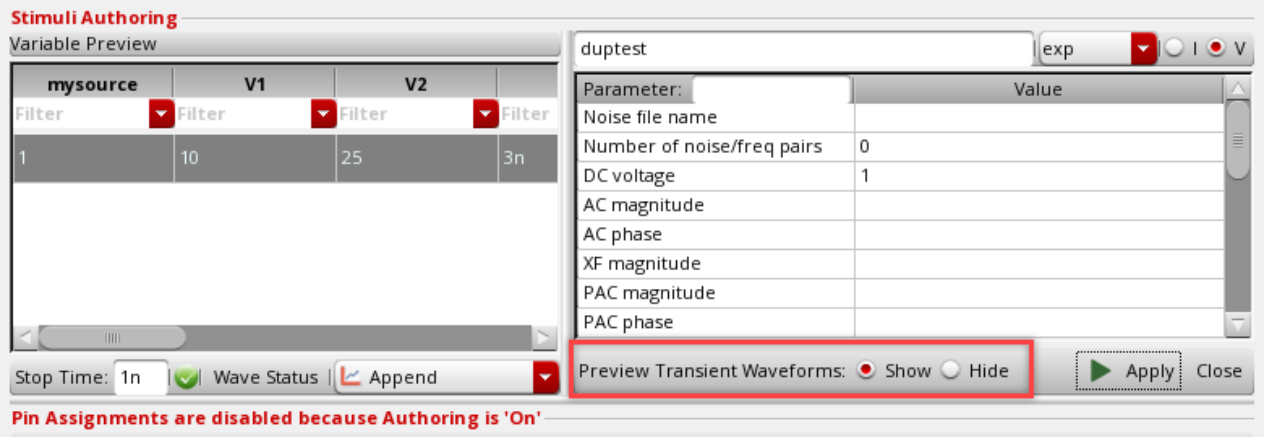

## Previewing Stimuli Waveform

To preview a stimulus created, follow the steps:

1. Ensure that the form is in authoring mode. To do this select *Authoring* `On`.


© 2023 Cadence Design Systems, Inc.

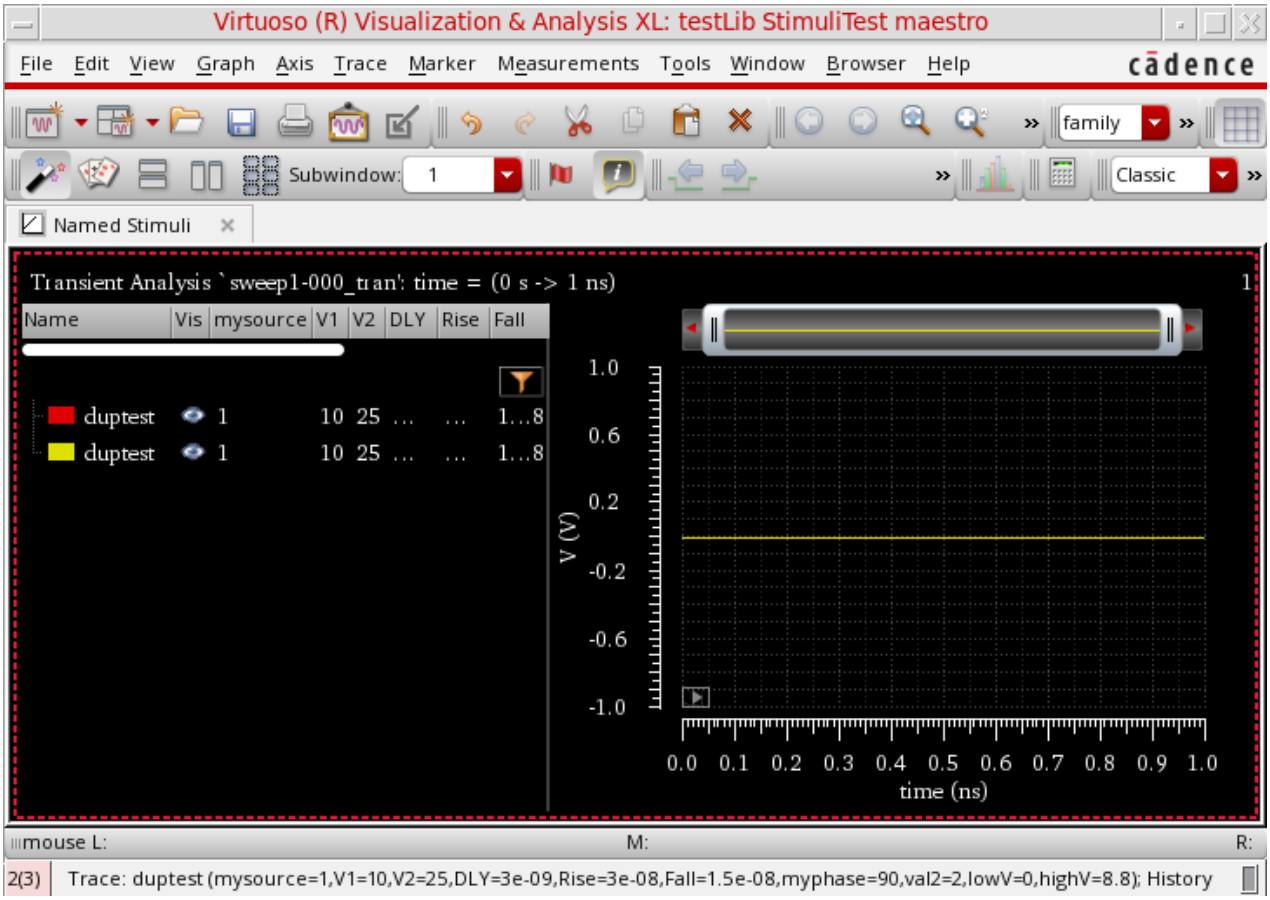
2. Select *Show* in *Preview Transient Waveforms.*
3. It opens the *Variable Preview* table that shows all the variables and their respective values.




4. Select one or more rows in the *Variable Preview* table and click Apply .


© 2023 Cadence Design Systems, Inc.

It generates a Virtuoso Visualization & Analysis XL preview of the selected stimulus.




Alternatively, you can use the function `maeStmGenerateWaveforms()` in CIW to generate waveforms in a batch mode for all the stimuli added to a maestro cellview.

### *Important Points*

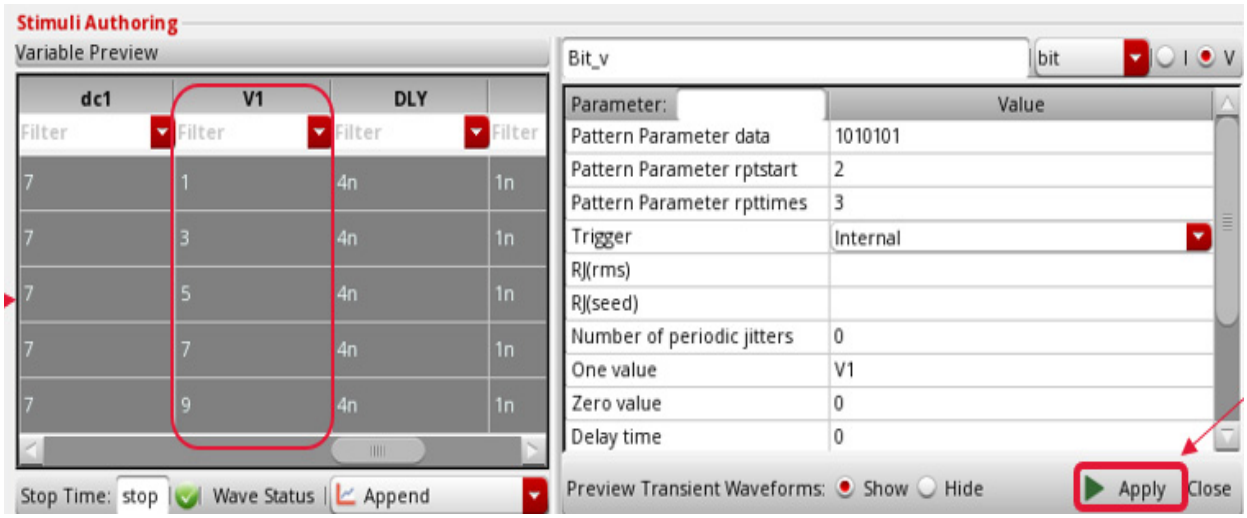
- The *Wave Status* shows a green tick

to indicate that the waveform is generated successfully for the specified stimuli.
- The Wave Status shows a warning sign

if the waveform for the specified stimuli is not generated.

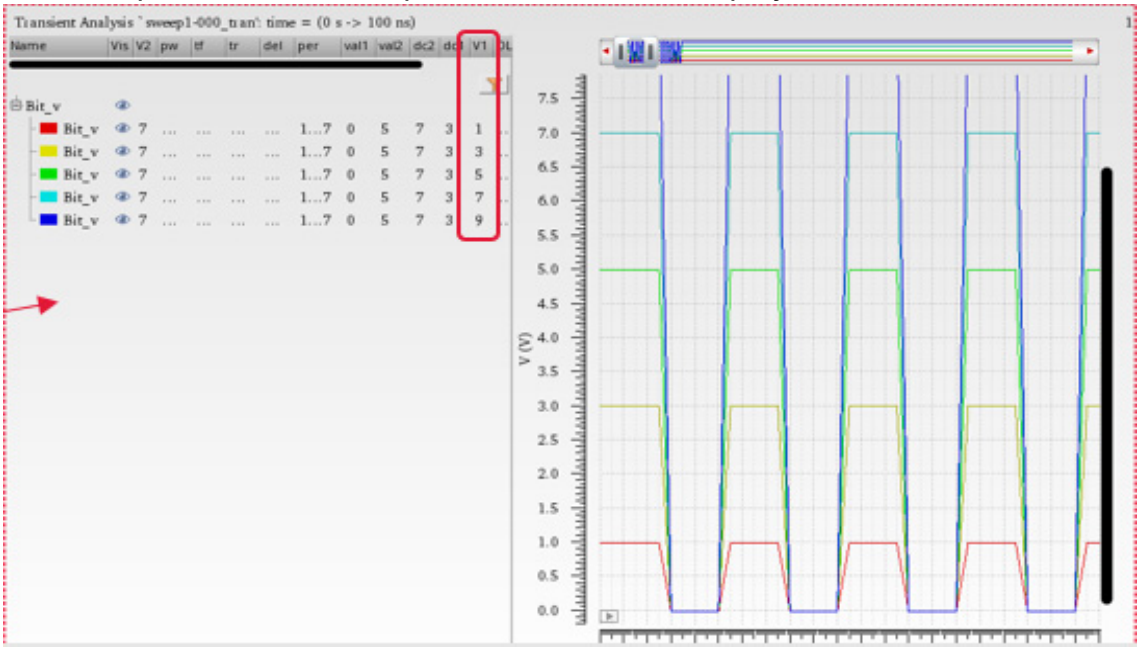

© 2023 Cadence Design Systems, Inc.

**Note:** You can select multiple sweeps of design variables defined in ADE Explorer. These are displayed, irrespective of whether they have been used in the stimulus definition or not.




If there are design variables in Explorer which are defined in quotes, such as "`aaaa`". They remain hidden from the *Variable Preview* table as quotes are not supported by the Simulator. The variables defined in quotes are not supported and are hidden from the table, to see these variables right click the header to select the columns or select to see all of them.

From the example above, the swept values of `V1` are displayed below.




If the Stimuli Authoring is in Off mode, you can select the stimuli and perform either of the following actions:


© 2023 Cadence Design Systems, Inc.

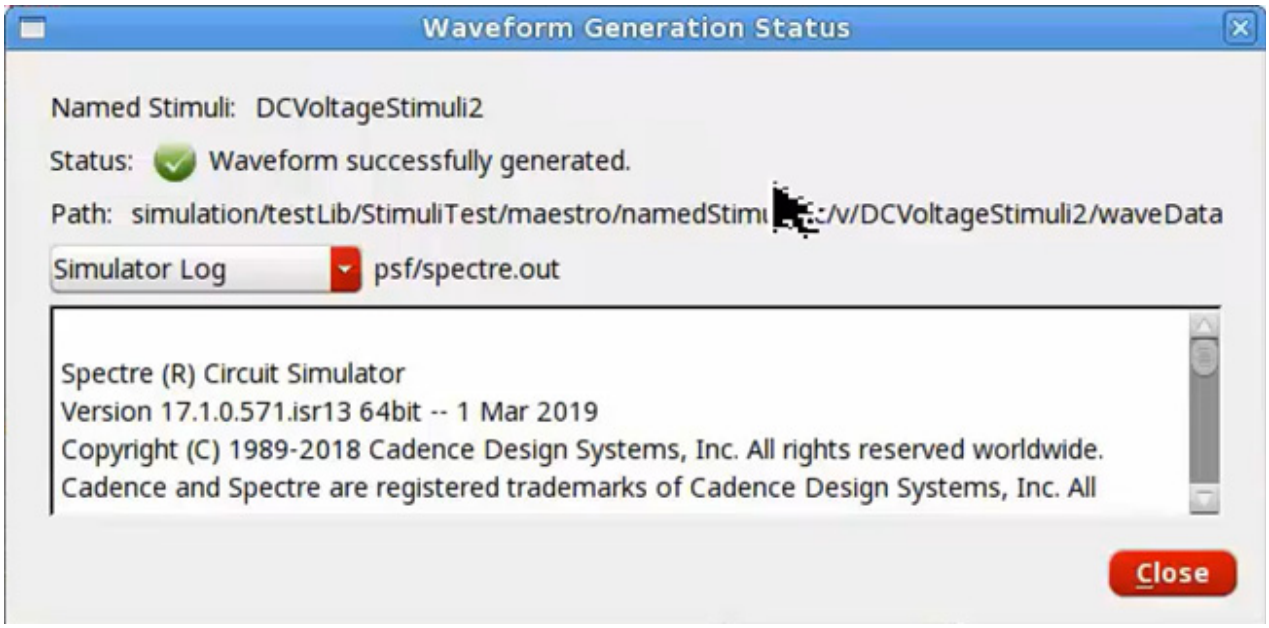
- Click *Plot* .
- Right-click in *Stimuli Authoring* section and select *Plot transient waveform*.

*Important*

The waveforms are saved in PSF format in the project directory.

### *Waveform Generation Status*

Click *Wave Status* in the bottom of the *Stimuli Authoring* section, to see the log file in the *Waveform Generation Status* message box.

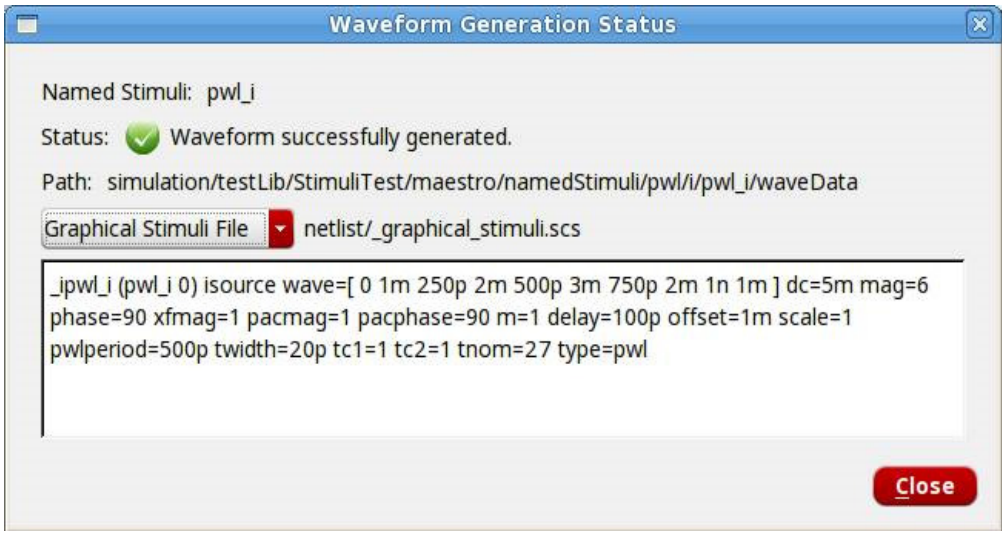



You can check information related to generating waveforms in the following ways.

- Simulator Log — Contains log information related to the simulator.

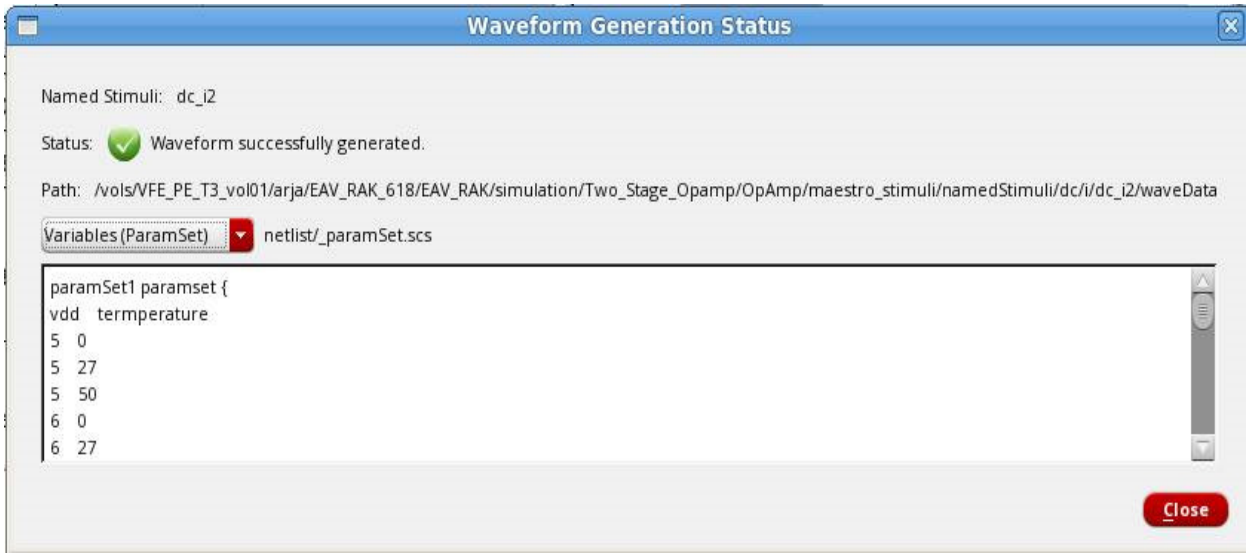

© 2023 Cadence Design Systems, Inc.

- Graphical Stimuli File — Contains stimuli definitions that are included in the netlist.




**Note:** The path to the netlist and psf directory is also shown in the *Waveform Generation Status*.

- Variables (ParamSet)— Contains the list of all sweeps points in the design variables defined in ADE Explorer.

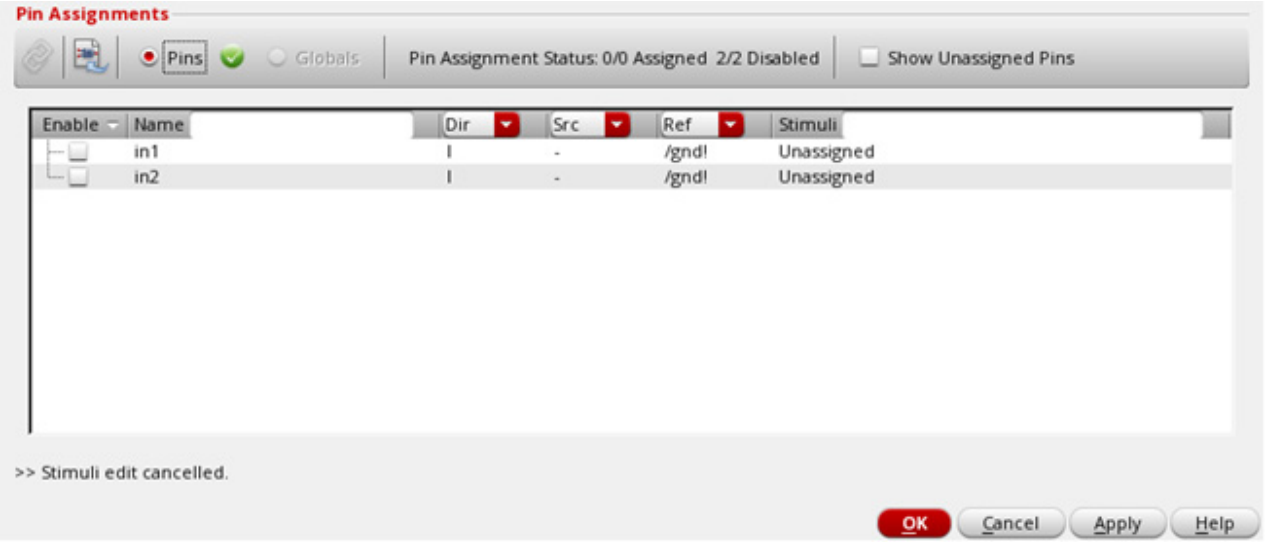



## Assigning Stimuli

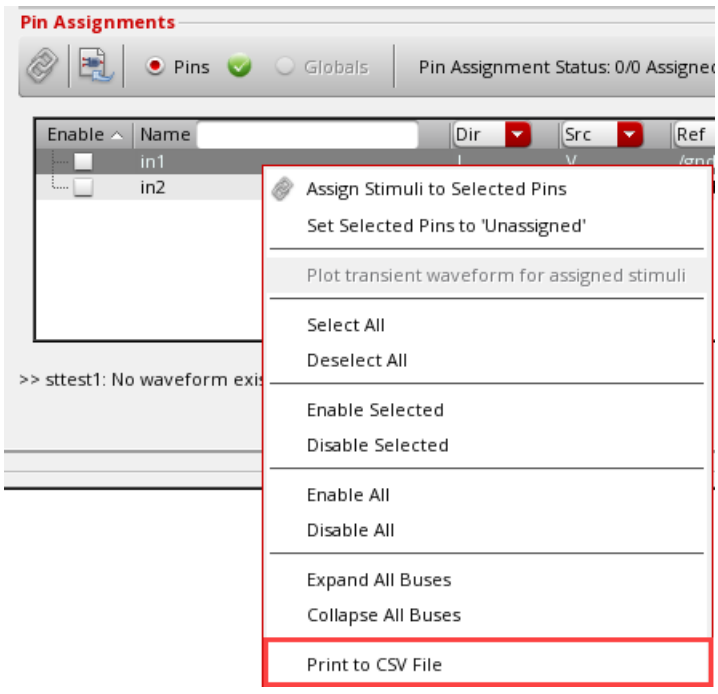
You can assign stimuli to pins, buses and global nets in the specified design. The *Stimuli Authoring* mode must be set to `Off` to assign stimuli or preview stimuli on pins, buses or nets.


© 2023 Cadence Design Systems, Inc.

All these design components are automatically fetched and are visible as a list in this section.You can run the simulation in ADE Explorer after pin assignments are completed.




To print the pin assignment table to a CSV file, right-click a pin name and select *Print to CSV File*.




The *Save Pin Assignments Table To CSV File* window is displayed.Either select an existing file or create a new one and specify its name in the *Field_name* field and click *Save*.

The contents of the pin assignments section will then be saved in that file in CSV format.


© 2023 Cadence Design Systems, Inc.

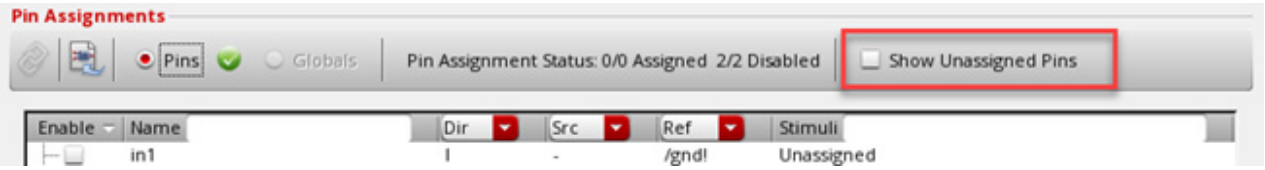
Assigning Stimuli to Pins

To assign a stimulus to a pin in a schematic, perform the following steps:

1. Select a stimulus from *Stimuli Authoring* section for an input signal.
2. Select an input pin in the *Pin Assignments* section.

*Tip*

You can select *Show Unassigned Pins* check box to filter to display the pins which do not have any stimuli applied to them.




3. Click .

   Alternatively, select the stimulus, and then, right-click the pin. Select *Assign Stimuli to Selected Pins* option.

   You can perform the above steps to assign stimuli to more than one pin.

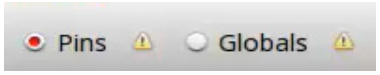
   **Note:** You can see the pin with an assigned stimulus in the schematic by selecting the pin and clicking .

   ***Important Points to note:***

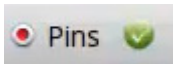
   - The *Pin Assignment Status* shows the `total number of pins assigned/ total number of pins in the design`. It also indicates the number of disabled pins. These numbers also include the bus bits in your design.

   - The warning icon near Pins indicates that there are enabled pins in the design which do not have assigned stimuli.
   - The warning icon near Globals indicates that there are enabled global sources in the design which do not have assigned stimuli.


© 2023 Cadence Design Systems, Inc.

- Once all pins are assigned stimuli, a green tick appears next to pins 

. This is applicable to global sources as well.

### *Assigning Stimuli to Global Sources*

To assign stimulus to global sources, select *Globals* and repeat the steps as explained in the section Assigning Stimuli to Pins.

**Note:** Global signals should be set to only DC sources.

### *Assigning Stimuli to Bus Bits*

You can either assign the same stimulus on all the bus bits or different stimuli on different bus bits.

To assign the same stimulus to bus bits:

1. Select a stimulus from the *Stimuli Authoring* section.
2. Select a bus in the *Pin Assignment* section.
3. Click .

**Note:** Press + to see all the children bus bits contained in a bus.

To assign different stimuli on the bus bits, perform the following steps for each bus bit:

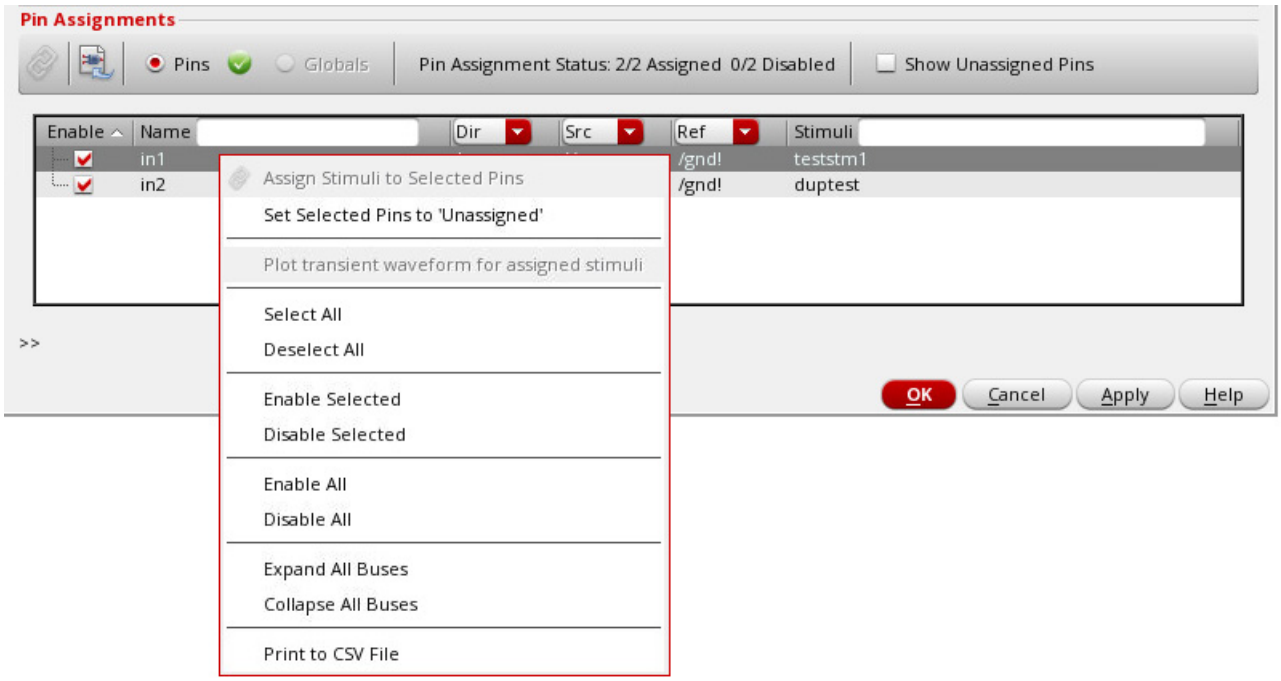
1. Select a stimulus from the *Stimuli Authoring* section.
2. Select the required bus bit in the *Pin Assignment* section.
3. Click .

Video

To know more, see the video, Creating a Bus from Stimuli Assignment Form that is available on Cadence Online Support


© 2023 Cadence Design Systems, Inc.

**Note:** To disable a bus bit, right-click the bus bit and select *Disable Selected*.

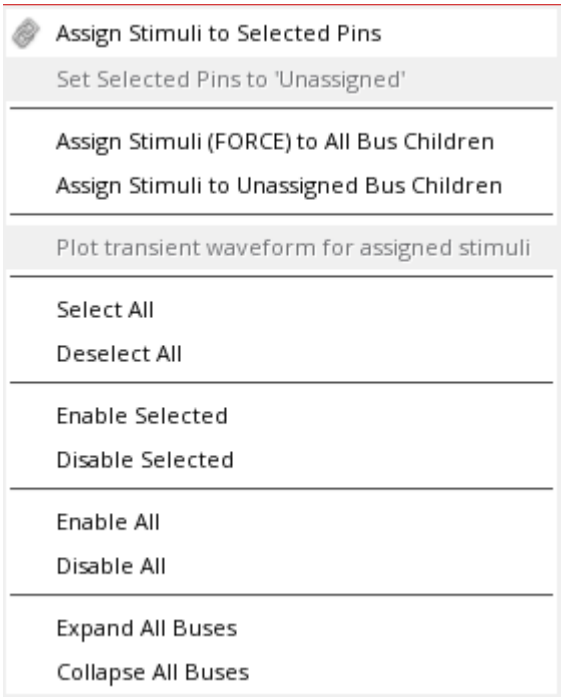




© 2023 Cadence Design Systems, Inc.

*Tip*

While you are working on pin assignments, you can use the commands in the context menu in the *Pin Assignments* section.




| Right-click Menu Option | Description |
|---|---|
| *Assign Stimuli (FORCE) to All Bus Children* | Forces the assignment of all bits of a bus to its children bits |
| *Assign Stimuli to Unassigned Bus Children* | Assigns all the unassigned bus bits |

Consider a scenario, if all the children buses are unassigned, you can choose the parent bus and select a stimulus. All the children bus bits will be assigned. But, if you do it again you will not be able to assign stimulus because the children bus bits in the parent bus have already been assigned a stimulus. In this case you need to select *Assign Stimuli (FORCE) to All Bus Children*. This overwrites the new stimulus assignment on the children bits.

In another scenario, if the bus is partially assigned, which means some children bits are assigned stimuli and the other bits are unassigned. You can choose a stimulus and the bus and select *Assign Stimuli to Unassigned Bus Children*. This way, the unassigned bits will


© 2023 Cadence Design Systems, Inc.

be assigned the selected stimulus. The previously assigned children bits continue to carry the old stimulus.

The description of stimuli status of parent bus bits:

- Unassigned–All its children bits are unassigned
- Partial–Some of its children bits are unassigned
- Various–All the children bits are assigned, but they are assigned different stimuli
- <Stimulus_Name>–All the children bits are assigned the same stimulus.

### *Plotting Waveforms for Assigned Stimuli*

To plot the waveforms for stimuli assigned to pins, bus, or bus bits, select all the pins in the *Pin Assignments* section. Right-click and select *Plot transient waveform for assigned stimuli*.

This enables a visual check on the stimuli assigned to each pin and bus bit in the design.

When you click *OK* on the Stimuli Assignment form, it checks whether all the enabled pins have stimuli assigned. If yes, it successfully saves it. Otherwise, it generates a warning message suggesting that you save and apply incomplete stimuli assignments in the design, which can lead to errors during netlist generation.

## Specifying a Stimulus File

Stimulus files let you add lines of code to the simulator input file that the analog circuit design environment generates. The stimulus file can be used to include input and power supply stimuli, initializing nodes, or including estimated parasitics in the netlist.

You can specify a stimulus file by using the Simulation Files Setup form.

## Example of a Spectre Stimulus File

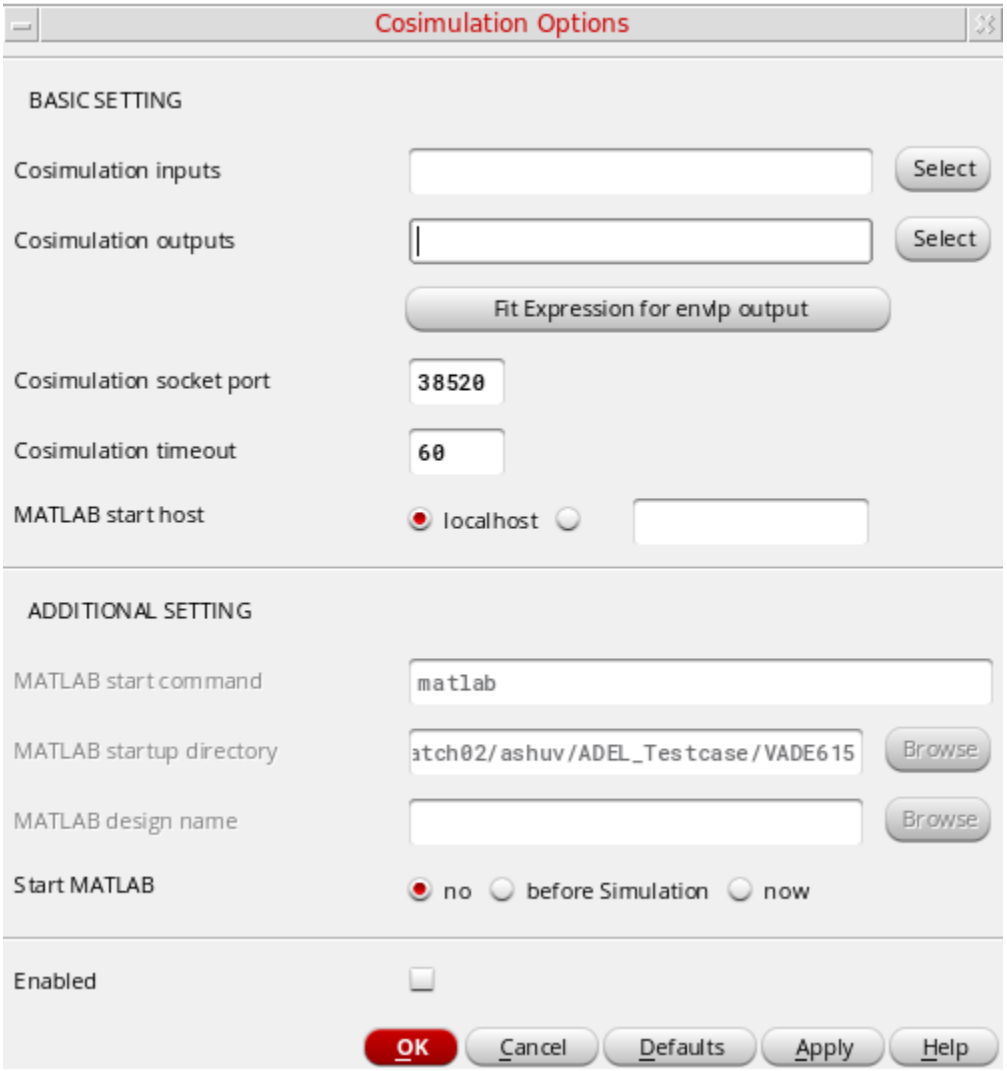
The file `opampStimuli.scs` in `tools/dfII/samples/artist/models/spectre` is an example of a stimulus file that can be used for the `opamp` example in the `aExample` library.

```
simulator lang=spectre

_v1 ([#inp] 0) type=sin freq=1k ampl=1
_v2 ([#inm] 0) type=dc dc=0
```


© 2023 Cadence Design Systems, Inc.

# Setting up Cosimulation Options

To set up the cosimulation options when the simulatior is spectre, choose *Setup – MATLAB/ Simulink – Setting*. The Cosimulation Options form appears.




1. In the *BASIC SETTING* section, specify the following fields:
   - In the *Cosimulation inputs* field, select the source instance from the schematic as input
   - In the *Cosimulation outputs* field, select the source a net or terminal from the schematic as output.
   - In the *Cosimulation socket port*, specify the socket port number.


© 2023 Cadence Design Systems, Inc.

- In the *Cosimulation timeout* field, specify the timeout value. The default value is 60 secs.
- In the *MATLAB start host* field, select the localhost option or specify a remote host name.

**2.** In the *ADDITIONAL SETTING* section, specify the following fields:

- In the *MATLAB start command* field, enter the command for starting MATLAB. The default command to start MATLAB is "`matlab`". You can enter any other command along with the command-line argument.
- In the *MATLAB startup directory* field, enter the path to the directory where matlab is launched in the
- In the *MATLAB design name* field, enter the path to the MATLAB design.
- Select one of the following options to start the Matlab:

| Select | If |
|---|---|
| *no* | If you select no option in Start MATLAB, MATLAB will not be started. This may be useful if you want to keep same form setup but do not want to run cosimulation or if you want to start MATLAB yourself, independent of ADE. |
| *before Simulation* | Select this option if you want to start MATLAB automatically, After you select this option, all the fields in the form are enabled. To run Simulink simulation automatically, you need to have a startup.m file that contains the sim() command in the directory, which is specified in the MATLAB startup directory field. |
| *now* | Select this option if you want to launch MATLAB manually. The Start button appears that you can click to launch MATLAB. |

**3.** Select the *Enabled* check box to enable cosiumation and click *OK*.


© 2023 Cadence Design Systems, Inc.

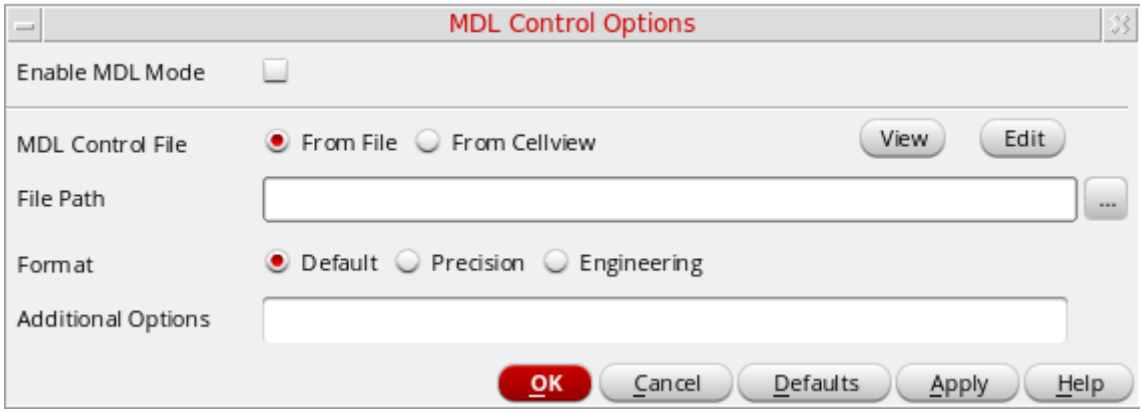
# Setting up MDL Mode

You can use the MDL mode to import an MDL control file in ADE Explorer environment. The MDL control file includes a script that defines measurements.

Perform the following steps to enable and setup the MDL mode:

1. Choose *Simulation — MDL Control*.

   The *MDL Control Options* form appears.




2. Select *Enable MDL Mode* to enable the MDL mode. This lets you add an MDL script that runs existing analyses defined in a netlist or that defines new analyses and measurements.

   When enabled, settings for all analyses are ignored.

3. In the *MDL Control File* field, select one of the following options to specify the source of the MDL file:

   - `From File`—When you select this option, the *File Path* field appears, in which you specify the file path.

     Alternatively, click the ellipses button to locate the file.

   - `From Cellview`—When you select this option, the *Library*, *Cell*, and *View* drop-down list boxes appear, in which you select the required library, cell, and view names.

     Alternatively, click the ellipses button to locate the view.

© 2023 Cadence Design Systems, Inc.

4. Click *View* to view the MDL file in read-only mode.
5. Click *Edit* to open the MDL file for editing.

   The MDL file opens in the default text editor set in Virtuoso.
6. In the *Format* field, choose one of the following formats to be used in calculations. The *Format* field specifies the notation style and default number of significant digits for the measured results.

   - `Default`—Displays the result in suffix notation.

     For example, `5.802m` is displayed as `5.802m` when number of significant digits is 4.
   - `Precision`—Displays the result in scientific notation and sets the maximum number of significant digits to be printed for `g`, `G`, `S`, and `s` types; or the number of decimal digits to be printed for `e`, `E`, and `f` types.

     For example, `%.15g` outputs the measured value to 15 significant digits.
   - `Engineering`—Displays the result in engineering notation and specifies the number of significant digits of the signal value output to the measurement file.

     For example, `6` outputs the signal value to six significant digits.
7. In the *Additional Options* field, specify the additional options.
8. Click *Default* to set all the options to back to their default values.
9. Click *Apply* to save the settings, and click *OK* to save the settings and close the form.

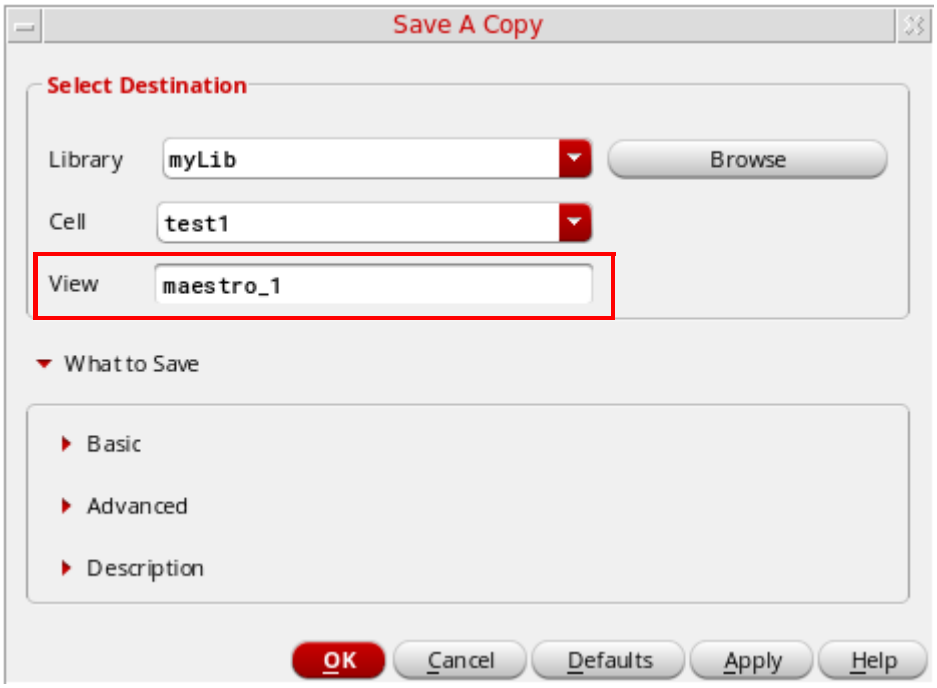
For more information, see *Virtuoso Simulator Measurement Description Language User Guide and reference*.


© 2023 Cadence Design Systems, Inc.

## Saving the Current Setup

To save the current ADE Explorer setup, do one of the following:

- *Session – Save*. The existing cellview is overwritten with the settings you defined in the current session.
- *Session – Save A Copy*. The Save A Copy form appears.




In the *Select Destination* section, select the following:

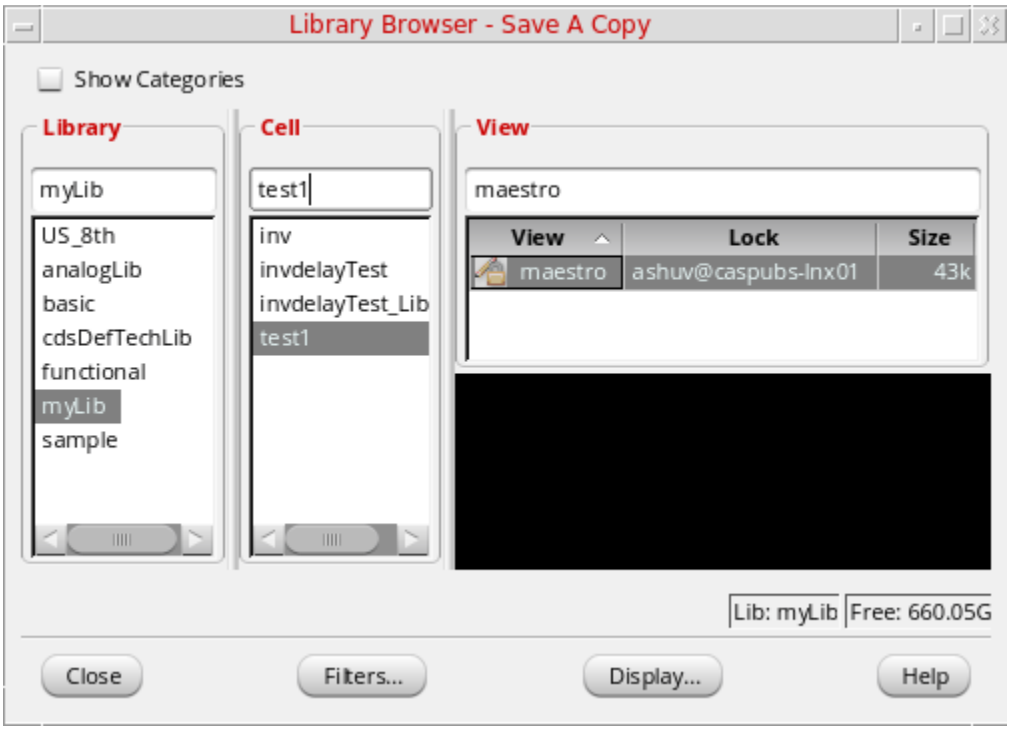
- In the *Library* field, select a library name to which you want to save the setup.
- In the *Cell* field, select a cell name from the specified library.
- In the *View* field, specify a view name in which you want to save the current ADE Explorer setup.

  By default, these fields display the library and cell that are currently opened in ADE Explorer. Notice that the new view name is similar to the current view name, such as `maestro_1`, as highlighted in the above figure. You can also specify a new view name.

  **Note:** Alternatively, click the *Browse* button to select the library, cell, and view using the Library Browser – Save a Copy form, as shown in the figure below. By default, the view


© 2023 Cadence Design Systems, Inc.

name changes to `maestro_n`.

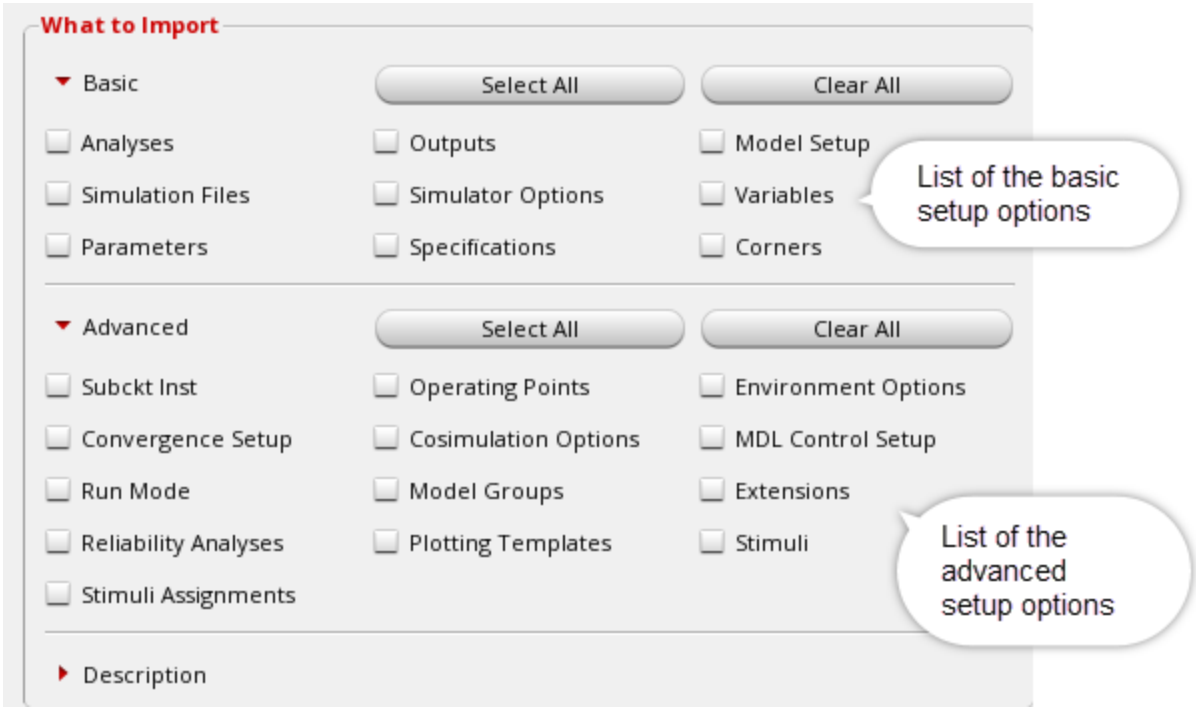



Click the *Close* button to close this form. The selected library, cell, and view are displayed in the respective fields in the Save a Copy form.

- In the *What to Save* section, do the following:
  - In the *Basic* sub-section, click the arrow symbol to view the basic setup options that you can save, such as Analyses, Outputs, Model Setup, and so on. To select all options, click the *Select All* button. To clear the selection of all options, click the *Clear All* button (see the figure below)
  - Similarly, in the *Advanced* sub-section, click the arrow symbol to view the advanced setup options that you can save, such as Subckt Inst, Operating Points, Environment Options, and so on. To select all options, click the *Select All* button. To clear the selection of all options, click the *Clear All* button (see the figure below).

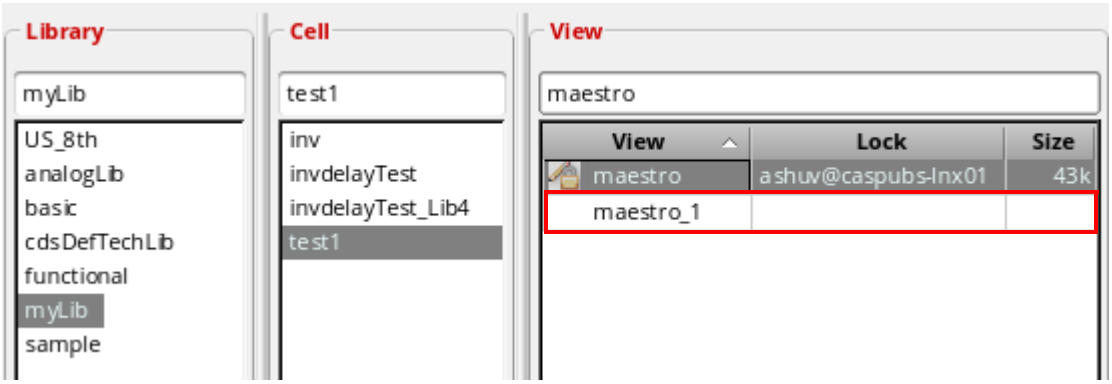

© 2023 Cadence Design Systems, Inc.

By default, all the options in the *Basic* and *Advanced* sub-sections are automatically selected when you specify a library, cell, and view. For more information about these options, see Load and Save Options on page 221.




- Expand the *Description* field and provide notes about the setup you are saving.
- Click *Apply* to save the specified setup and then click *OK* to close this form.

  The selected design and settings are saved in the specified cellview. For example, the figure below shows that a new cellview, `maestro_1`, is added for the specified library and cell in the Library Manager.





© 2023 Cadence Design Systems, Inc.

## Importing a Saved Setup

You can import the setup in ADE Explorer that you had previously saved using ADE Explorer or Assembler. In this case, you do not need to close the current ADE Explorer window to open a new setup and you can directly import another existing setup in the same window.

**Note:** Unlike ADE Assembler, the design and simulator settings cannot be imported in ADE Explorer.

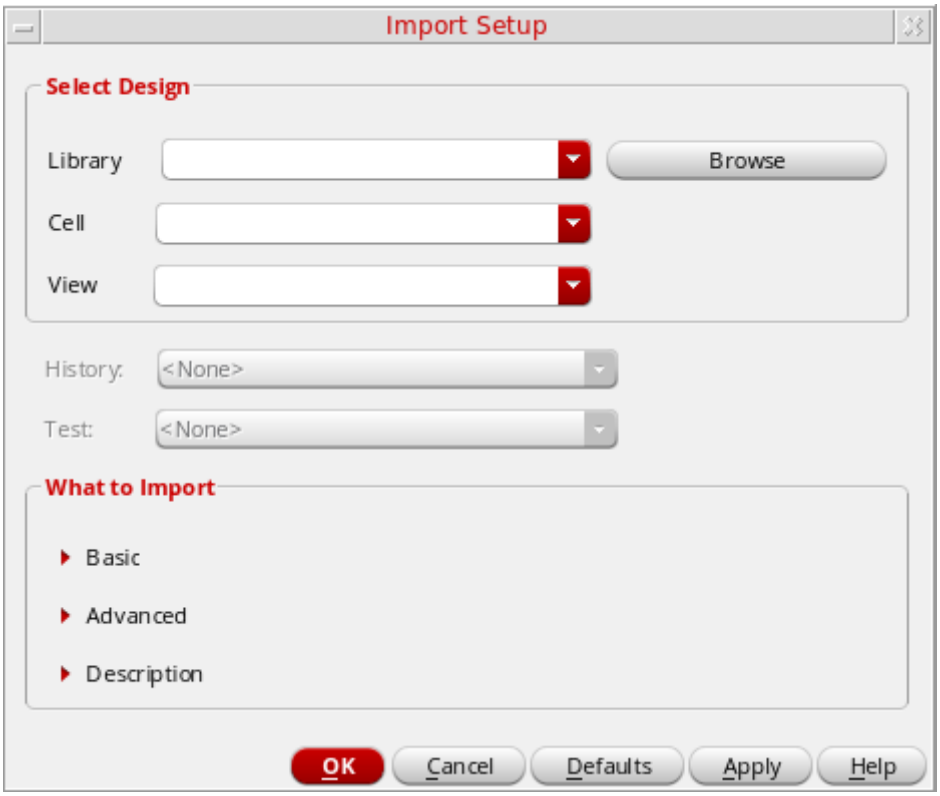
Video

The video Saving and Importing a Setup is available on Cadence Online Support demonstrates how to save the current setup in ADE Explorer and how to import a saved setup in ADE Explorer.

To import a previously saved setup:

➡ Choose *Session – Import*.

The Import Setup form appears.

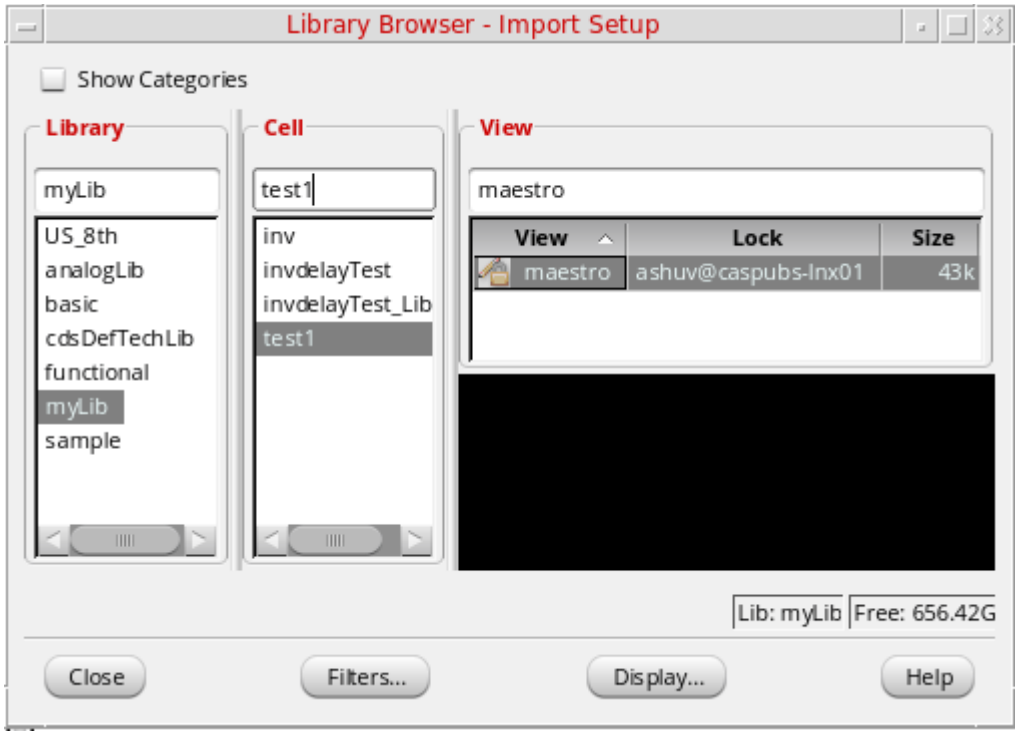



In the *Select Design* section, select the following:


© 2023 Cadence Design Systems, Inc.

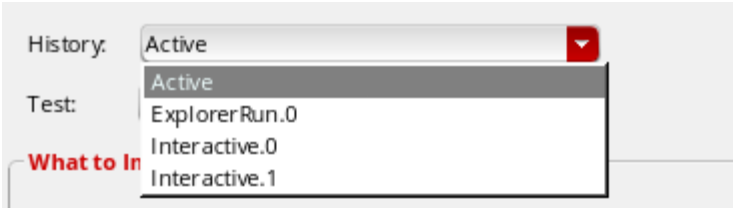
- In the *Library* field, select a library name from which you want to import the setup.
- In the *Cell* field, select a cell name from the specified library.
- In the *View* field, select a view name that you want to import.

  **Note:** Alternatively, click the *Browse* button to select the library, cell, and view using the Library Browser – Import Setup form, as shown in the figure below. Only views of type `maestro` are displayed in the Library Manager.




  Click the *Close* button to close this form. The selected library, cell, and view are displayed in the respective fields in the Import Setup form.

- In the *History* drop-down list, select a saved setup history that you want to import. The `Active` history indicates the current setup. This field becomes active only when the library, cell, and view information is selected in the form.





© 2023 Cadence Design Systems, Inc.

- In the *Test* drop-down list, select the test that you want to import. Multiple tests are available when the `maestro` cellview is created using ADE Assembler. When the selected design does not have a test available for importing, a warning message appears.

  When you have multiple tests available, only components specific to the selected test are overwritten, while the components that are applicable for other tests and the current maestro view remain unchanged. If the setup includes conflicting corners, they will be uniquely renamed. For example, suppose you have two maestro views:

  - `maestro_1`, with tests `T1` and `T2`. `T1` includes a corner, `C1`, `T2` includes a corner, `C2`.
  - `maestro_2`, with test `T1` and corner `C1`. Notice corner `C1` exists in both the views.

  Now, if you import `maestro_2/T1` into `maestro_1/T1`, the corner `C1` in `maestro_1/T1` is overwritten by value in the imported test. Notice the value of corner `C2` is unchanged because the imported test does not have this corner.

  Now, suppose you have another maestro view, `maestro_3`, which has test `T1` and corner `C2`, and you import this view into `maestro_1/Test1`. `The maestro_1` view already includes a corner `C2` for test `T2`, but this corner is not visible in the ADE Explorer setup because currently opened test is `T1`. Therefore, the corner `C2` is renamed to `C2_0` in the imported view.

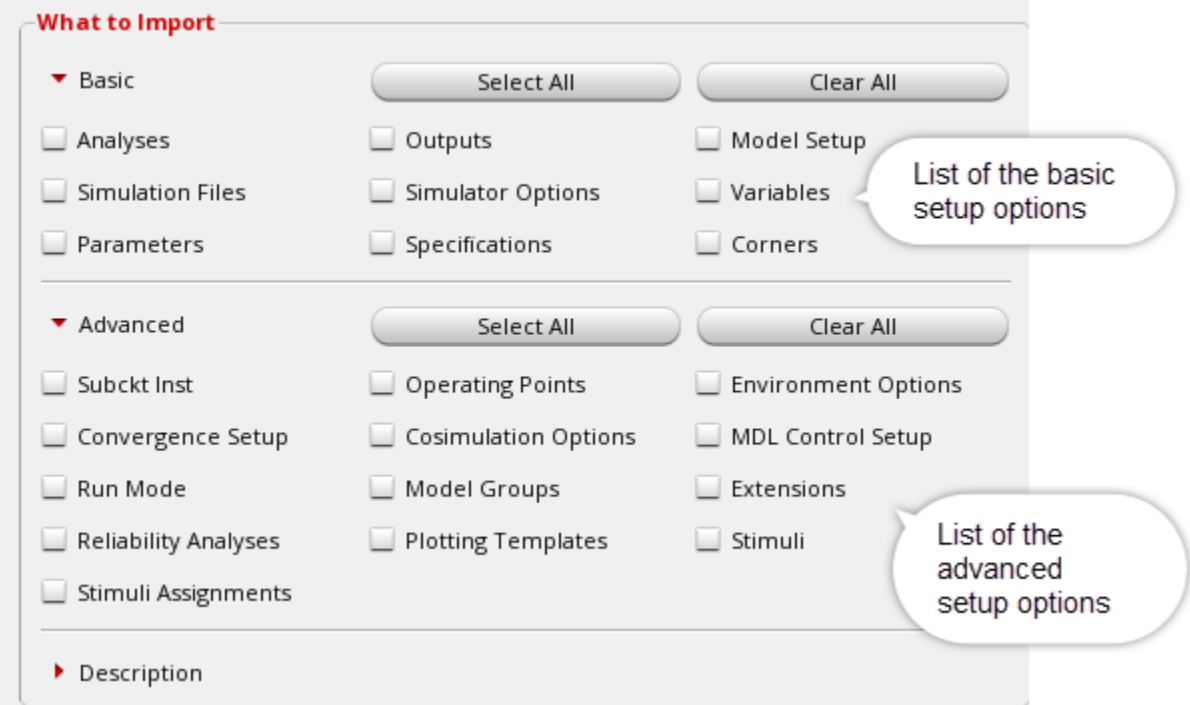
- In the *What to Import* section, do the following:

  - In the *Basic* sub-section, click the arrow symbol to view the basic setup options that you can import, such as Analyses, Outputs, Model Setup, and so on. To select all options, click the *Select All* button. To clear the selection of all options, click the *Clear All* button (see the figure below)
  - Similarly, in the *Advanced* sub-section, click the arrow symbol to view the advanced setup options that you can import, such as Subckt Inst, Operating Points, Environment Options, and so on. To select all options, click the *Select All* button. To clear the selection of all options, click the *Clear All* button (see the figure below)

  **Note:** By default, all the options in the *Basic* and *Advanced* sub-sections are selected when you specify a library, cell, and view. If an option is not available for the given setup, it is disabled in the form. For more information about these options, see Load and Save


© 2023 Cadence Design Systems, Inc.

Options on page 221.




- Expand the *Description* field to view or add any notes, if available, for the selected view.
- Click *Apply* and then click *OK* to close this form.

  The selected settings are imported in ADE Explorer and can be viewed in the Setup assistant.

  **Note:** When the current view in ADE Explorer includes a configuration view, the setting of *Multi-Technology Mode* option (MTS) on Choosing Simulator form is imported implicitly. If the current view is not a configuration view, an error message is displayed in CIW.

*Important*

If you have any environment variable settings specified for the options selected in the Import Setup form, those settings will be overridden by the settings specified for them in the cellview that you want to import.

### Load and Save Options

The following options are available on the What to Import group box:


© 2023 Cadence Design Systems, Inc.

### *Basic*

The following table describes the basic options in the Import Setup form for Spectre and AMS simulators.

| Option | Description |
|---|---|
| *Analyses* | Restores or saves the *Choosing Analyses* form settings (but not simulator options that you set with the Options buttons in these forms). Analysis settings are simulator specific, so you cannot restore them from a different simulator. |
| *Outputs* | Restores or saves the saved and plotted output settings. Output settings are design specific, but if the signal names match, you can restore them from a different design. You can also restore expressions from any design. |
| *Model Setup* | Restores or saves the model setup of the session that was saved. |
| *Simulation Files* | Restores or saves simulation-specific files and other information. For Spectre, these files can be stimulus files, definition files, and include paths. |
| *Simulator Options* | Restores or saves all the settings from the Simulator Options form. This information is simulator specific. |
| *Device Checking Setup* | Restores or saves all the settings from the Device Checking Options form. |
| *High-Performance Simulation* | Restores or saves the settings from the High-Performance Simulation Options form. |
| *Variables* | Restores or saves design variables. |
| *Parameters* | Restores or saves parameters. |
| *Specifications* | Restores or saves specifications from the Outputs Setup pane. |
| *Corners* | Restores or saves corners defined in the Corner Setup form. |


© 2023 Cadence Design Systems, Inc.

### *Advanced Options*

The following table describes the advanced options in the Import Setup form for both Spectre and AMS simulators.

| Option | Description |
|---|---|
| *Subckt Inst* | Restores or saves the settings specified in the *Save By Subckt* tab of the Save Options form. |
| *Operating Points* | Restores or saves operating points outputs specified in the Outputs Setup pane. |
| *Environment Options* | Restores or saves only the environment option settings. If you are using a different simulator in another session, you can restore only those options that are applicable to the current simulator. |
| *Convergence Setup* | Restores or saves the *Node Set*, *Initial Condition*, and *Force Node* settings, as well as DC and transient solutions. |
| *EMIR Options* | Restores or saves the settings from the EMIR Analysis Setup form. |
| *MTS Options* | Restores or saves the settings from the MTS Options form. |
| *Run Mode* | Restores or saves the specified run mode and run options. The run modes can be Single Run, Sweeps and Corners or Monte Carlo. |
| *Model Groups* | Restores or saves the model groups specified in the Model Library form |
| *Extensions* | Restores or saves the High Performance simulation options settings. This information is simulator-specific. |
| *Reliability Analyses* | Restores or saves the setup of the Reliability analyses options form. |
| *Plotting Templates* | Restores or saves the settings from the Plotting Template form. |
| *Stimuli Definitions* | Restores or saves the stimuli definitions specified in the *Stimuli Authoring* group box of the Stimuli Assignment form. |
| *Stimuli Assignments* | Restores or saves the stimuli assignments specified in the *Pin Assignments* group box of the Stimuli Assignment form. |


© 2023 Cadence Design Systems, Inc.

The following table describes the advanced options that are specific to Spectre simulator.

| Option | Description |
|---|---|
| *Cosimulation Options* | Restores or saves the Matlab/Simulink cosimulation options settings. |
| *MDL Control Setup* | Restores or saves the MDL control options, which runs Spectre with the MDL control file. |

The following table describes the additional advanced options that are specific to AMS simulator.

| Option | Description |
|---|---|
| *Solver Option* | Restores or saves the solver specification for the current session. |
| *Run Option* | Restores or saves the run option setting as batch or interactive as specified in the Choose Run Options form. |
| *MATLAB and Simulink* | Restores or saves the settings for MATLAB/Simulink co-simulation. |
| *Connect Rules/IE Setup* | Restores or saves Restores or saves connect rules and related information about modules, disciplines and resolution as specified in the Connect Rules Selection form.<br>**Note:** To import the connect rules, ensure that the E*nvironment Options* check box is selected as well. |
| *Discipline Section* | Restores or saves disciplines as specified in the Default Digital Discipline Selection form. |
| *Global Signals* | Restores or saves global signals specified through the Global Signals form. |
| *Library Files* | Restores or saves the library file and directory specifications. |

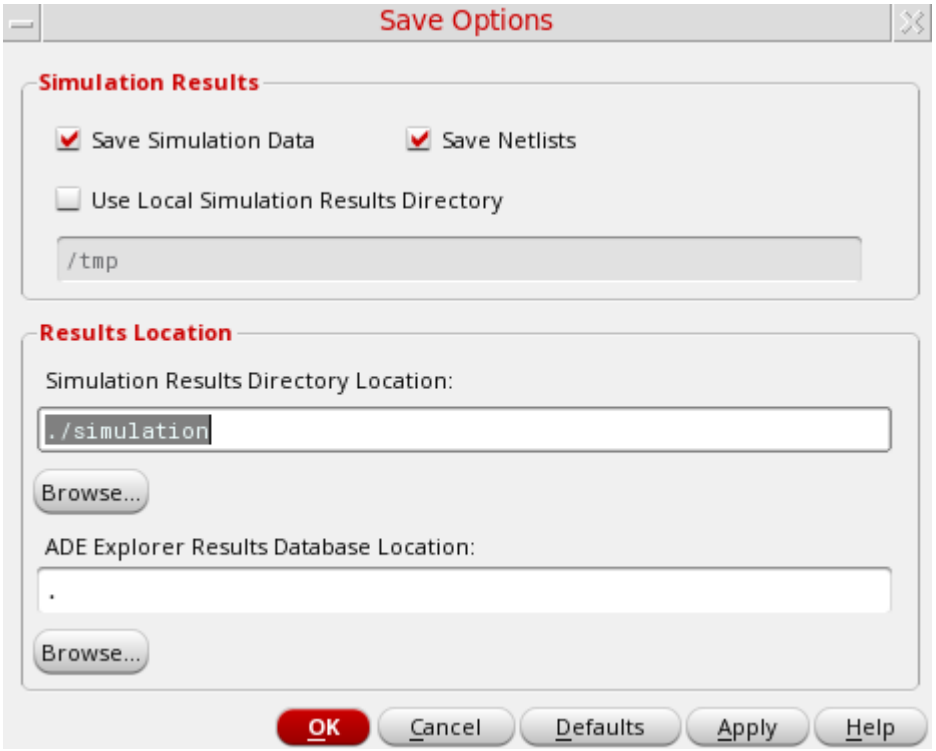

© 2023 Cadence Design Systems, Inc.

## Specifying Simulation Results Directory Location

To specify the results directory location where simulation results are to be saved:

➡ Choose *Setup – Save Options*.

The *Save Options* form appears, as shown in the figure below.




■ In the *Simulation Results* section, you can specify which simulation data must be saved. Specify the following fields:

- ❑ Select the *Save Simulation Data* check box to save the simulation data results in the specified results directory. This check box is selected by default.
- ❑ Select the *Save Netlist* check box to save the generated netlist at the specified results location. This check box is selected by default.

**Note:** By default, the results for distributed simulation runs are saved in the location specified in the *Simulation Results Directory Location* field.

- ❑ Select the *Use Local Simulation Results Directory* check box to save the results for distributed simulation jobs run on a remote system in a local directory. Specify the path to the local directory in the subsequent field.


© 2023 Cadence Design Systems, Inc.

For example, if you set the local simulation results directory as `/tmp`, all the simulation data is saved in the remote host's `/tmp` directory.

**Note:** While using remote or LSF job distributions, the local GUI host cannot directly access the remote host's local directory. This means, the plotting and simulation results become inaccessible. You should select this option only if you do not want to access these results.

- In the *Results Location* section, specify the following fields:
  - In the *Simulation Results Directory Location* field, specify the location where you want to save the simulation results. By default, the results are saved in the `~/simulation` directory in the current directory path.
  - In the *ADE Explorer Results Database Location* field, specify the location where you want to save the results directory.

***Related Topics***

saveEvaluationErrorResults

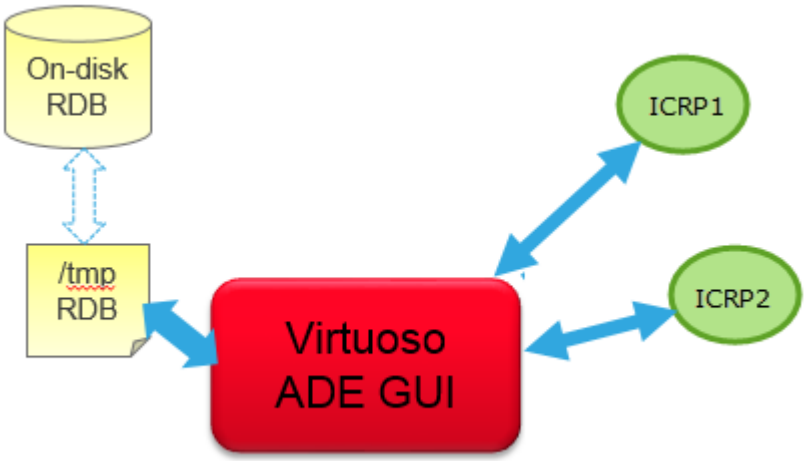
## How does ADE Explorer save the results in the results database?

Prior to IC6.1.7 ISR10, ADE Explorer and the IC remote processes (ICRP) used to write to the .rdb file on the file system, which is an NFS disk. However, the NFS file locks, which are used to ensure database correctness, are slow. This used to impact the performance resulting in a non-responsive GUI during simulation and a lengthy simulation run time. In addition, simultaneous attempts to access the `.rdb` file by the ADE Explorer user interface and the ICRP increase the possibility of database corruption due to the unreliability of NFS file locking mechanism.

To avoid this, in IC6.1.7 ISR10, ADE Explorer uses a local directory, named as `/tmp`, to temporarily save the results database. This avoids the NFS file lock issues and centralizes access to the `.rdb` file because it can be accessed only through the ADE Explorer UI. In addition, it uses a separate backup thread to copy the results data from the /tmp directory to the permanent results database, the location specified by the *ADE Explorer Results Database Location* field.


© 2023 Cadence Design Systems, Inc.

ADE Explorer saves a .rdb file in the `libraryName/cellName/viewName/results/data directory` under the specified location.




### Important Points to Note

ADE Explorer saves the `.rdb` file to the `/tmp` directory on the same computer where the application is running.

- Only the ADE Explorer user interface can access `/tmp`, therefore, it is not required to be locked and released for every read or write operation, and database operations are also fast.
- If Virtuoso is closed abruptly due to some issue and all the data has not been copied to the results database, all the settings for block transfer and frequency of backup are ignored, and all the unsaved data is copied from the `/tmp` directory to the results database.
- The availability of disk space for both `/tmp` directory and the `.rdb` file is checked periodically. If the available disk space becomes less than the lower limit, appropriate warning messages are flagged. Also, if these disks do not have sufficient space, the simulation is automatically suspended. You can free up the space and resume the simulation.
- When you close the ADE Explorer window, it runs a check to confirm that all the data from `/tmp` directory has been successfully copied to the `.rdb` results database. If some data remains to be copied, ADE Explorer waits before closing the window, and shows the following message in the CIW.


© 2023 Cadence Design Systems, Inc.

```
ADE is in the process of copying the simulation results from the /tmp directory,
which is created to provide faster access to the data, to a permanent location.
This may take some time depending on the size of data to be copied. It is
recommended not to forcefully close the application as it can result into data
loss.
```

- You can choose to save the results database directly in the .rdb file on the NFS disk instead of /tmp directory by setting the `RDBCreatePolicy` environment variable to "`NFSDisk`". By default, this variable is set to "`TmpDir`".


© 2023 Cadence Design Systems, Inc.

# Saving Results in ADE Explorer

By default, in edit mode, the Virtuoso ADE Explorer saves the results of only the recent simulation run with the default history filename, `ExplorerRun.0` at the following location:

```
<simulationDir>/<lib>/<cell>/<view>/results/maestro/ExplorerRun.0/
```

Every time a new simulation is run, the `ExplorerRun.0` results file is overwritten to store the results from the latest simulation run. However, you can also save on-demand results (histories) in ADE Explorer that can be loaded later.

When the ADE Explorer is opened in a read-only mode, the `ExplorerRORun.0.RO` history file is created at the following location:

```
<simulationDir>/<lib>/<cell>/<view>/results/maestro/
ExplorerRORun.0.RO/
```

*Important*

You can simultaneously open two or more ADE Explorer sessions in read-only mode using the same `maestro` cellview. These sessions share the same ADE Explorer history item, `ExplorerRORun.0.RO`. Therefore, you cannot run two simulations in parallel because both the simulations need write access to the same history item. You can run simulations in a sequence, one after the other. The most-recent simulation run will overwrite the history file `ExplorerRORun.0.RO`.
If you want to save the results obtained from both the simulations, then save the results of the runs individually, or if you want to run simulations simultaneously, switch to ADE Assembler.
When the `ExplorerRun.0` or `ExplorerRORun.0.RO` histories already exist and you try to run a simulation in ADE Explorer, an error message is displayed mentioning that the simulation cannot be run. To run the simulation in this case, you need to delete these histories by selecting them from the *Results – Delete* menu.

*Important*

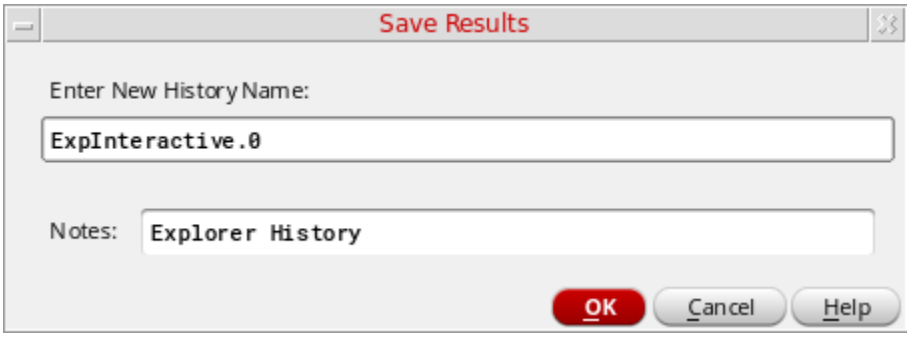
If a history is opened in multiple ADE Explorer sessions, and you rename or delete that history in one session, the history is closed in all the sessions except the one in which the change is initiated. Similarly, if you run a simulation to overwrite a history, ADE Explorer closes the history from other sessions, if found open.

Follow one of the below steps to save the results in ADE Explorer after the simulation run is complete:

- Choose *Results – Save*.


© 2023 Cadence Design Systems, Inc.

- Right-click in the *Value* column of the Outputs Setup pane.The *Save Results* form appears as shown in the figure below.




In this form, do the following

1. In the *Enter New History Name* field, specify a history name to store the simulation results of the current simulation.

   **Note:** For single point, multipoint run, or corners, the default history name is `ExpInteractive.0` and for Monte Carlo simulation, the default history name is `ExpMonteCarlo.0`.

2. In the *Notes:* field, specify a description about this history.

3. Click *OK*.

   The history of the current simulation run is saved with the specified name, `TestHistory`.

   If you save the same Explorer setup with a new history name without running the simulation, the previously-specified history file is overwritten. For example, if you save the same setup saved in the `TestHistory`, with a new name `TestHistory1`, the `TestHistory` is renamed to `TestHistory1`.

   After running a single point simulation and exiting the session, if you re-open the `maestro` view in Explorer, or descend to the same test from Assembler, the previous set of results are automatically loaded.

   When you specify an already existing history name to save the results, the following pop-up message is displayed:

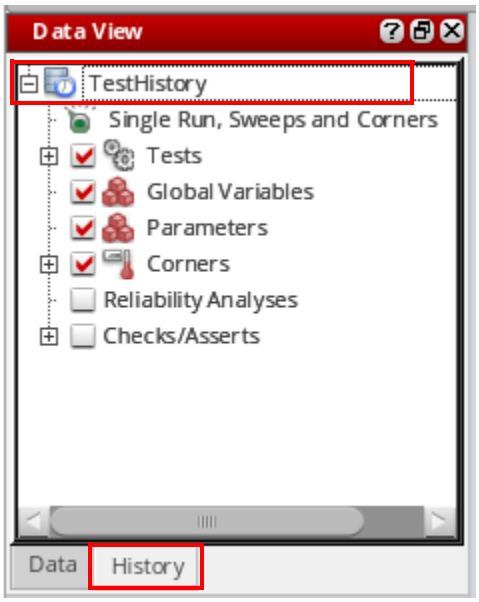
   ```
   Cannot modify history '%s' with value '%s' because this history item already
   exists. Do you want to overwrite this history item?
   ```

   By default, the maximum of 10 histories can be saved in ADE Explorer. However, when you load histories saved in ADE Assembler into ADE Explorer, the ADE Explorer histories are also retained even if the total history count goes beyond the specified limit.


© 2023 Cadence Design Systems, Inc.

To load results from a multi-point run, open the `maestro` view and choose *Results – View – ExplorerRun.0*.

The results you save in Explorer are visible on the *History* tab in Assembler, as shown in the figure below:



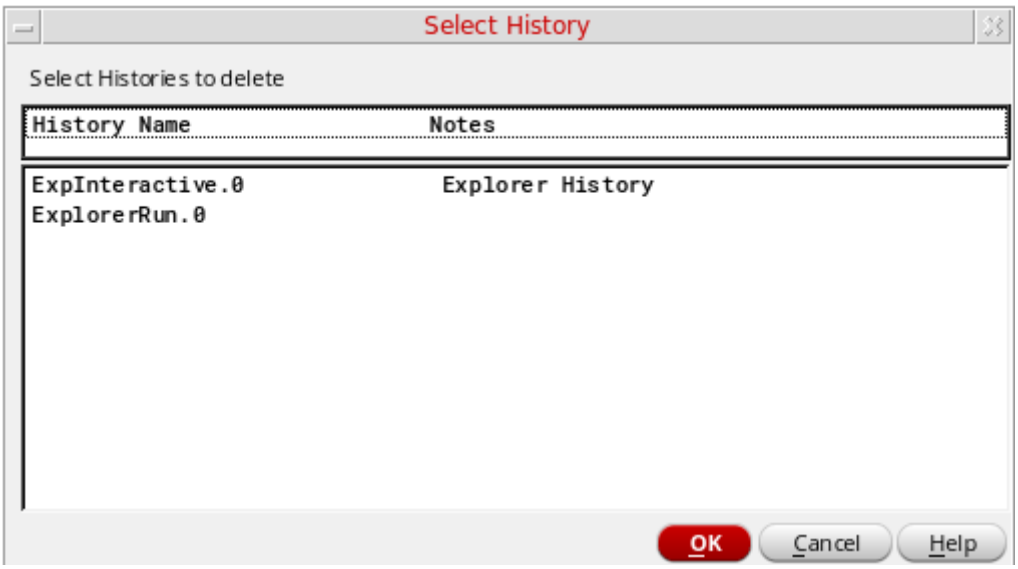

After running a single-point simulation, If you close the ADE Explorer session and then open the session again, the simulation results and scalar values are automatically loaded in the Value column of the Outputs Setup pane. However, if you re-open a session in which you ran multi-point simulation previously, the results are not automatically loaded because of their heavy size. To view or perform plotting operations on those results, you can choose *Results – View – ExplorerRun.0*.

### Deleting Saved Results

To delete a previously-saved history, perform the following steps:


© 2023 Cadence Design Systems, Inc.

1. Choose *Results – Delete.* The *Select History* form appears as shown in the below figure. This form lists all the previously-saved histories.




2. Select the history name that you want to delete.
3. Click *OK*.

### Viewing Saved Results

To view a saved history,

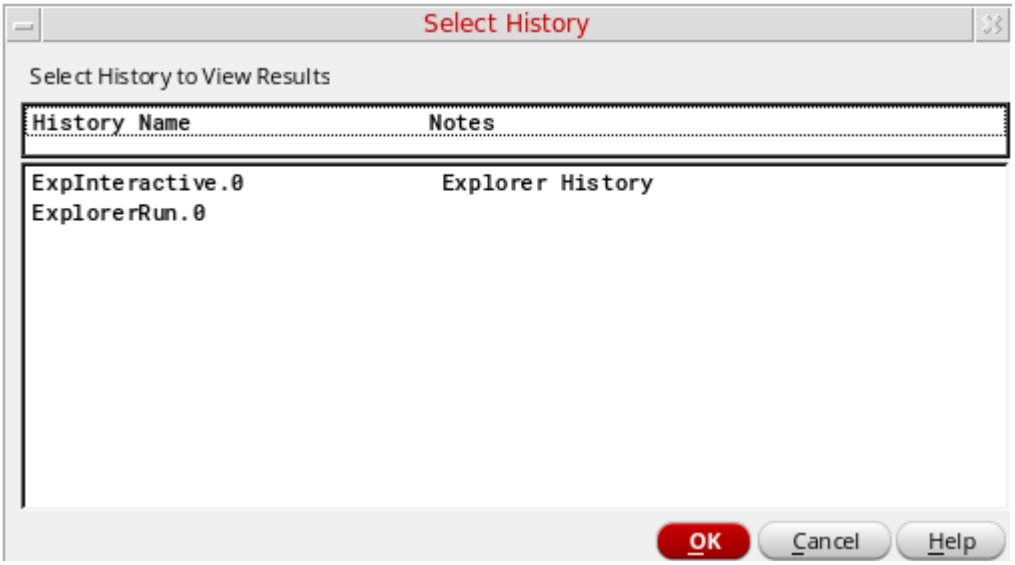
➡ Choose *Results – Select.*

The Select History form appears. In this form, in the *History Name* column, select the history you want to view the results for. Any notes added for a history item are displayed in the *Notes* column.

**Note:** If no history is available in the current session, the *Result – Select* option is disabled. If multiple histories are available, by default the first history is selected in the


© 2023 Cadence Design Systems, Inc.

history list.




The results from the selected simulation run history are displayed in the outputs setups

**Note:** The results of the selected history must be compatible with the existing setup. For a single point history run, if the selected history has different outputs, the incompatible outputs will not be displayed.


© 2023 Cadence Design Systems, Inc.

# Saving an OCEAN Script

The Open Command Environment for Analysis (OCEAN) helps you set up, simulate, and analyze circuit data. OCEAN is a text-based process that you can run from a UNIX shell or from the Command Interpreter Window (CIW). You can type in OCEAN commands in an interactive session, or you can create scripts containing your commands, then load those scripts into OCEAN. OCEAN can be used with any simulator integrated into the analog circuit design environment.

The OCEAN script helps you do the following:

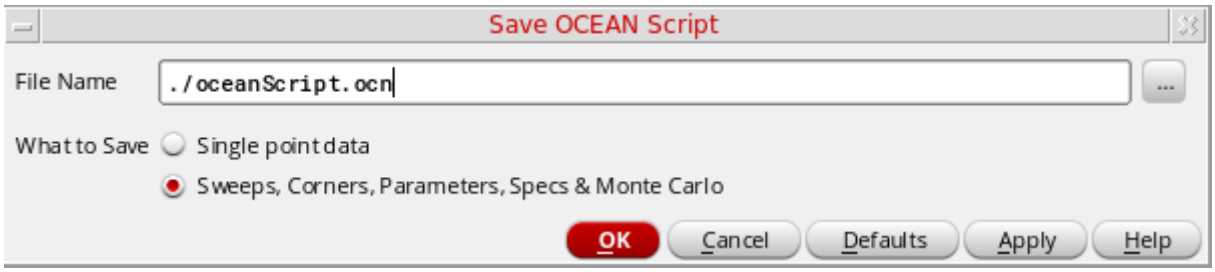
- Create scripts that you can run repeatedly to verify circuit performance
- Run longer analyses such as parametric analyses, corners analyses, and Monte Carlo simulation, more effectively
- Run long simulations in OCEAN without using the graphical user interface
- Run simulations from a nongraphic, remote terminal

In ADE Explorer, you can set up the simulation environment, choose plotting options, and then save these settings in a script.

To save an OCEAN script:

1. Create the setup and plot processing selections that you want to capture in the script.
2. Choose *Session – Save Ocean Script*

   The *Save OCEAN Script* form appears.




   The OCEAN XL script starts with `ocnxlSetXLMode()` and saves simulation commands for corners and Monte Carlo. The OCEAN L script does not save any XL related information even if the ADE Explorer setup contains corners or Monte Carlo. In addition, when an OCEAN XL script is saved using ADE Explorer, the simulation run modes that are not supported in ADE Explorer are not written to the script.

   **Note:** When you switch from ADE Assembler to ADE Explorer, the OCEAN script shows


© 2023 Cadence Design Systems, Inc.

the setup information of only the selected test.

3. In the *File Name* field, specify the name of the OCEAN file to be saved. By default, it displays the `./oceanScript.ocn` filename.
4. In the `What to Save` field, select one of the following options:
   - `Single point data` if you want to save an OCEAN L script that includes only single-point data.
   - `Sweeps, Corners, Parameters, Specs & Monte Carlo` to save an OCEAN XL script. This option also saves the local design variable sweeps.
5. Click *OK*.

   ADE Explorer creates and displays a script containing the OCEAN setup and plotting tasks you selected. You can edit the script to add simulation or postprocessing commands as needed.

   Alternatively, you can use a text editor to create an OCEAN script by using OCEAN commands and save it in a `.ocn` file.

   **Note:** For a single point run, ADE Explorer can run multiple simulations simultaneously using different OCEAN sessions.

   **Note:** The *`spectreFormatter`* and associated methods are defined in the *spectreinl* context. This does not get loaded automatically by *asiGetTool('spectre*). To load relevant contexts, you need to edit your code by adding *spectreinl* to the list of contexts it loads. This is illustrated in the following example:

```
; Need to have these contexts loaded - in the right order
; - so that that the spectreFormatter class can be extended
(foreach context '("oasis" "asimenv" "analog" "spectrei" "spectreinl")
  (unless (isContextLoaded context)
          (loadContext
           (prependInstallPath (strcat "etc/context/" context ".cxt")))
          (callInitProc context)
          )
  )
```

To know more about how to run an OCEAN script and how to run parallel OCEAN XL simulation runs for a maestro View, see Creating and Running an OCEAN Script in *Virtuoso ADE Assembler User Guide*.

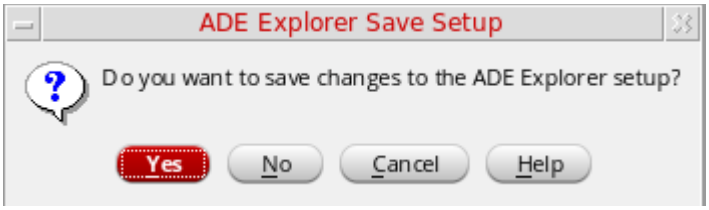
For more information about OCEAN commands and scripts, see the *OCEAN Reference*.


© 2023 Cadence Design Systems, Inc.

## Closing a Session

To close an ADE Explorer session, choose *Session – Quit*.

The following form appears:




Click *Yes* to save the current setup of the ADE Explorer. Click *No* to close the session without saving setup or click *Cancel* to continue working in the current setup.


© 2023 Cadence Design Systems, Inc.

# Configuring and Customizing ADE Explorer Settings

The configuration of ADE Explorer depends on the following types of configuration settings:

- Settings in your personal and site `.cdsinit` files
- Settings in your personal and site `.cdsenv` files
- UNIX environment variables in your `.cshrc` file

## Customizing Your .cdsinit File

You can customize your analog circuit design environment by adding SKILL commands to your `.cdsinit` file, the initialization file for the Cadence software.

There is a sample `.cdsinit` file for the analog circuit design environment in the following location:

*your_install_dir*`/tools/dfII/samples/artist/.cdsinit`

## Customizing Your .cdsenv File

You can set the default values for fields in analog circuit design environment forms by setting variables in your `~/.cdsenv` file.

There is a sample `.cdsenv` file for the analog circuit design environment in the following location:

*your_install_dir*`/tools/dfII/etc/tools/`*simulator_name*

## Customizing Your Menus File

The menus file is a simple SKILL file, therefore you can customize the same menu file for different releases by adding SKILL code within the *if-then-else* statement.

```
(if (equal curVersion 44X) then
   ;;; 44X menus customization here
else
   ;;; 44Y menus customization here
)
```

Alternatively, you can create the `simui.menus` file like this:

```
(if (equal curVersion 44X) then
    load "44X_file"
```


© 2023 Cadence Design Systems, Inc.

```
)
(if (equal curVersion 44X) then
    load "44Y_file"
)
```

where `44X_file` and `44Y_file` are path to the menus file for different releases.

## Setting UNIX Environment Variables

UNIX environment variables help configure the Virtuoso® analog design simulation environment.

The `CDS_Netlisting_Mode` variable controls

- How parameter values on components that use CDF and AEL are interpreted during netlisting
- Which LVS tool the system uses

The syntax for this variable in a `.cshrc` file is

```
setenv CDS_Netlisting_Mode "{Analog|Digital|Compatibility}"}
```

When the `CDS_Netlisting_Mode` variable is set to `Analog` or `Compatibility`, the component parameter evaluation takes CDF and AEL into account. When the variable is set to `Digital`, CDF and AEL are not taken into account, which results in better netlisting performance. When the variable is set to `Analog`, the Analog LVS tool (`auLvs`) is used.

Use these rules to set `CDS_Netlisting_Mode`.

- When you use the analog circuit design environment, set this variable to an appropriate value. If your design depends on CDF information, then set `CDS_Netlisting_Mode` to `Analog`. If your circuit does not use CDF information, then set `CDS_Netlisting_Mode` to `Digital` or `Compatibility`.
- When you use socket simulation in the analog circuit design environment and this variable is not set or is set to `Digital`, you see a dialog box that lets you set the value to `Analog` for the current session. (The socket netlister requires that the variable be set to `Analog`.) In this situation, the design environment uses the Analog LVS (`auLVS`) tool.
- If you use the analog circuit design environment for LVS, set `CDS_Netlisting_Mode` to `Analog`.
- If you use CDF and the Circuit Designer's Workbench but do not have the analog circuit design environment, set `CDS_Netlisting_Mode` to `Analog`.


© 2023 Cadence Design Systems, Inc.

- If you do not have the `auLVS` tool and do not use CDF or AEL expressions, set the variable to `Digital`. In this mode, you get the fastest netlisting speed and run `iLVS`, which uses OSS NLP expression syntax.
- If you want CDF compatibility with `iLVS`, set the variable to `Compatibility` for faster netlisting than with `Analog`. However, the `Compatibility` mode has the following limitations:
  - Connection of terminals by properties is not supported. A typical use of this capability in the analog circuit design environment is connecting the bulk pin of four-terminal transistors to a net.
  - Analog circuit design environment features other than expression evaluation are not supported.

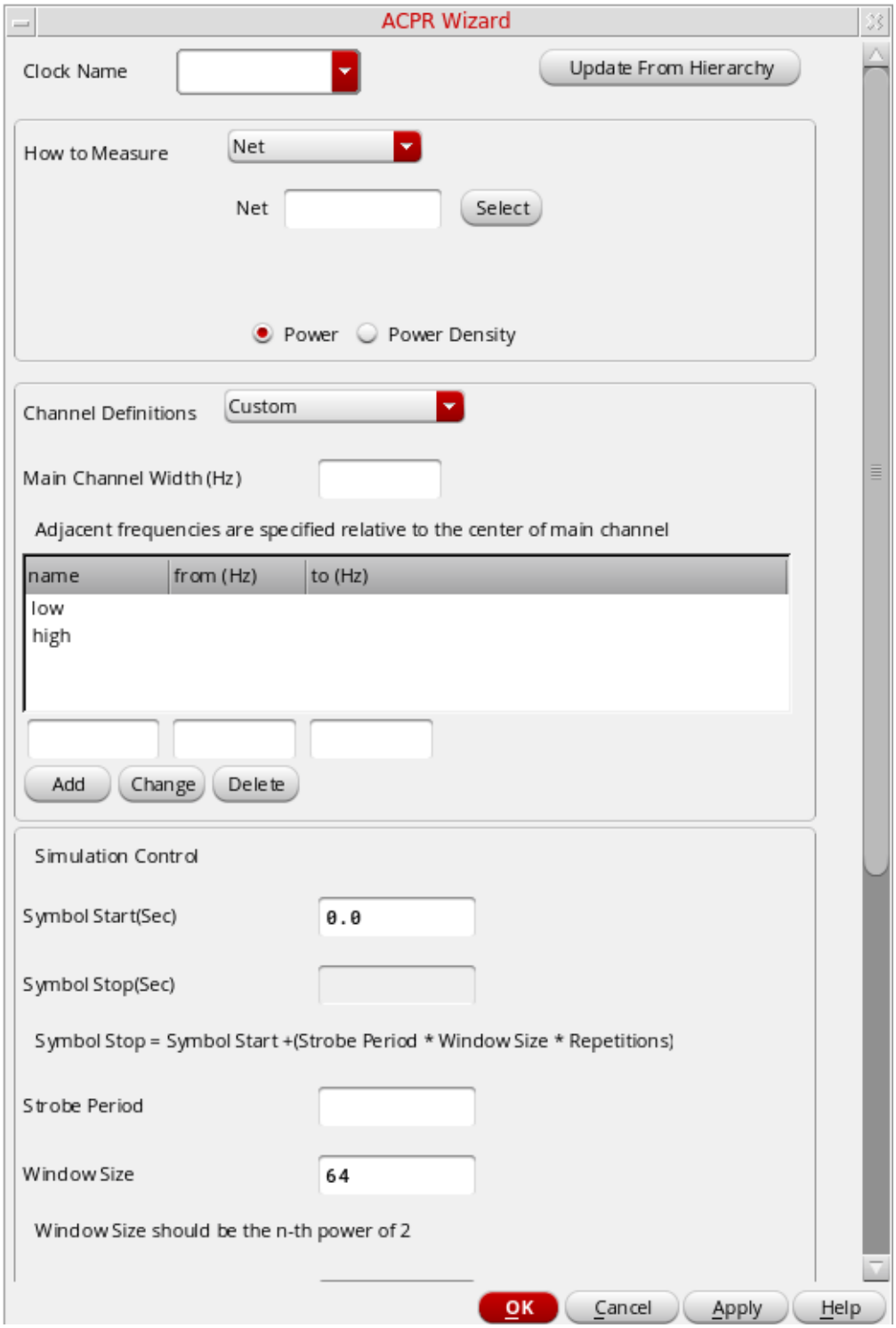
**Note:** By default, `CDS_Netlisting_Mode` variable is set to `Analog`.

## RF Options

When the simulator is set to Spectre, do the following to set up the ACPR (Adjacent Channel Power Ratio) and LSSP (Large Signal S-Parameter) options:


© 2023 Cadence Design Systems, Inc.

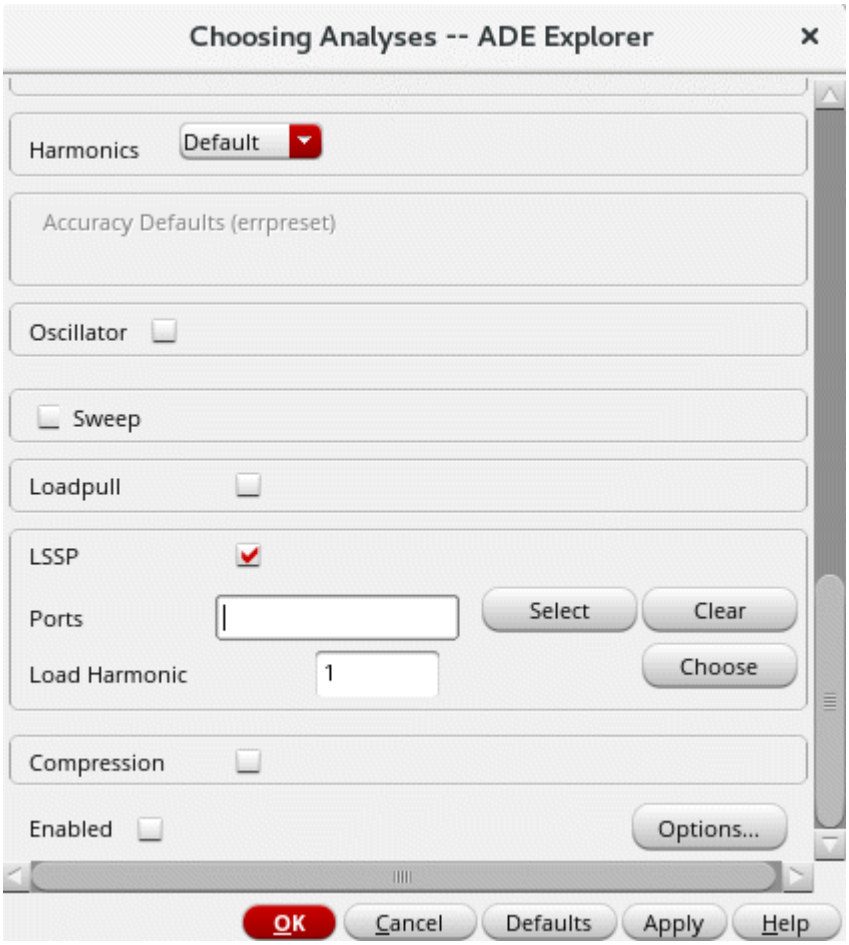
- Choose *Tools – RF – ACPR.* The ACPR Wizard form appears.





© 2023 Cadence Design Systems, Inc.

- Select the *LSSP* check box in the Choosing Analysis form for harmonic balance analysis.




For more information about the fields on these forms, see *Spectre Circuit SImulator and Accelerated Parallel Simulator RF User Guide* and *Spectre Circuit SImulator RF Analysis Theory*.


© 2023 Cadence Design Systems, Inc.

# Using Signals or Triggers

The ADE Explorer session functions are a known set of states. Any transition from one state to another is called an event. You can specify customized actions to be automatically performed whenever an event occurs. You can do this by registering callbacks for these events or signals. In Trolltech QT's terminology, these events are known as signals and the callbacks are known as slots.

To execute callbacks or slots for ADE Explorer signals, you need to perform the following tasks:

1. Define a callback function

   Define any SKILL function that you need to call when an event occurs in ADE Explorer. It is recommended to define this procedure in the .cdsinit file.

2. You can use any other SKILL function in this callback function.

   Connect the defined callback function with the signal

   Connect the custom SKILL function defined in step 1 with the required signal or event by using the `axlSessionConnect` SKILL function in the .cdsinit file.

3. Register the callback when a ADE (G)XL session is launched

   To make the callback function available for calling, it is required to register the function by using the axlSessionRegisterCreationCallback SKILL function in the .cdsinit file.

*Tip*

To know the signature of a trigger, you can use the axlSessionSignalSignature SKILL function.

For more information about how to set the triggers, see the examples in the Working with ADE (G)XL Signals or Triggers section of *Virtuoso ADE XL SKILL Reference*.


© 2023 Cadence Design Systems, Inc.

# Reserved Words

Each simulator has reserved words you cannot use as names for design variables or passed parameters of a subcircuit (use in a pPar expression):

- Simulator command or function names
- Simulator global variables, including
  - Simulator options
  - Names on fixed-form fields

    For example, CDF parameters on CDF forms, or properties on property forms, are all reserved words.

All Spectre simulator reserved words can be found in the *Creating Component and Node Names* section of the *Spectre Circuit Simulator Reference.*


© 2023 Cadence Design Systems, Inc.

© 2023 Cadence Design Systems, Inc.

# 3

# Setting Up Analyses

This chapter demonstrates how to set up in order to run an analysis.


- Setting Up Analyses
- Specifying Order for Analyses
- Setting Up a Spectre Analysis
- Setting Up an Analysis for UltraSim
- Setting Up an Analysis for AMS
- Setting Up an Analysis for HSPICE
- Setting Up EM/IR Analysis



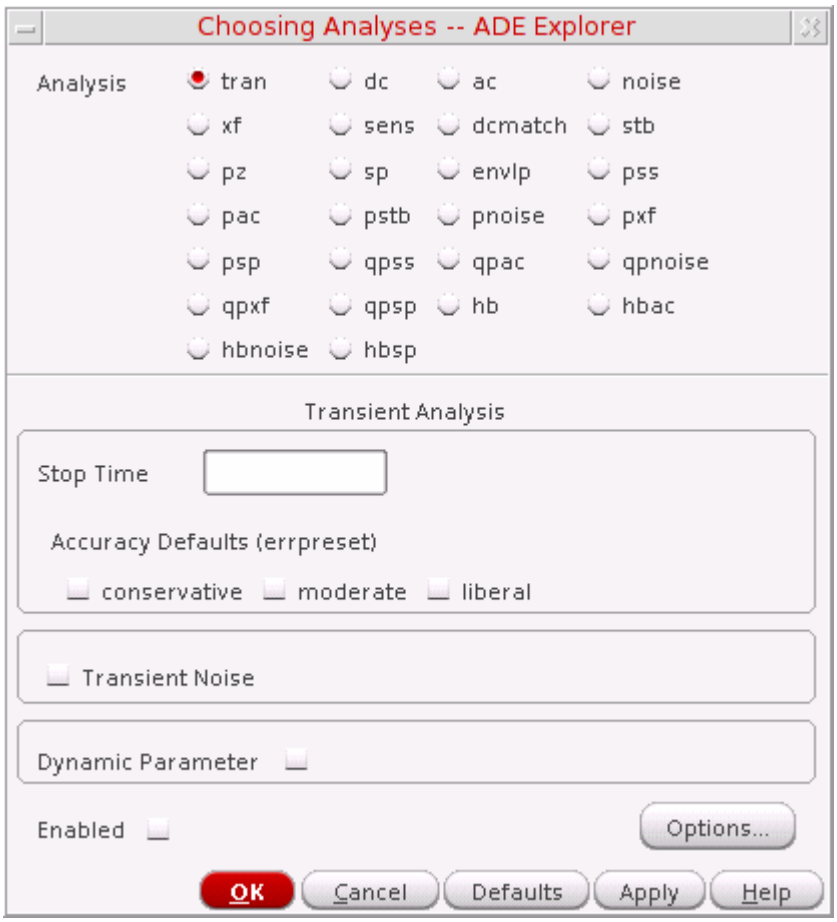
© 2023 Cadence Design Systems, Inc.

# Setting Up Analyses

Perform one of the following steps to set up an analysis:

- Choose *Analyses – Choose*.
- Click the *Click to add analysis* text in the *Analysis* section of the Setup assistant.
- Click the Choose Analysis button .

  The *Choosing Analyses* form for the specified simulator appears.




In this form, perform the following steps:

**a.** Select an analysis.

**b.** The *Choosing Analyses* form is updated to show the parameters for the selected analysis.

**c.** Set the analysis options.

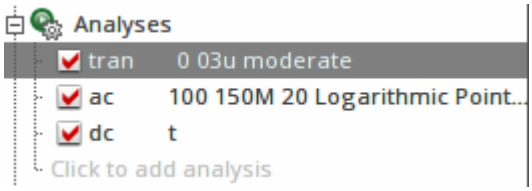

© 2023 Cadence Design Systems, Inc.

**d.** Select the *Enabled* check box to enable the analysis.

**e.** Click *Apply*.

**Note:** When you click *Apply*, the settings for all the enabled analyses specified through the form are saved.

The selected analyses get displayed in the Setup assistant of the ADE Explorer window.




For more information about how to load the existing analyses using the *Loading State* form, see Setting up MDL Mode on page 213.

## Disabling an Analysis

Perform the following steps to temporarily disable an analysis without deleting it:

1. In the Setup assistant, select the analysis to highlight it.
2. Choose *Analyses – Disable*.

**Note:** You can also use the *Enable* check box in the Setup assistant of the ADE Explorer window to enable or disable an analysis.

## Deleting an Analysis

Perform the following steps to delete an analysis:

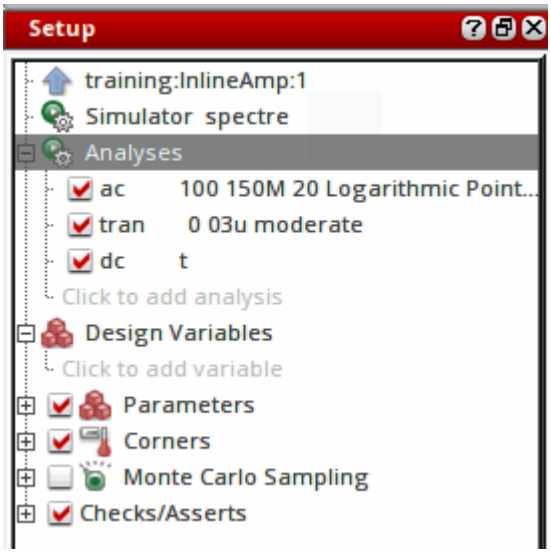
1. In the Setup assistant, click an analysis to highlight it and choose *Analyses – Delete* or click the delete icon.
2. Alternatively, in the Setup assistant, right-click an analysis and choose *Delete*.

For help regarding setting up a particular analysis, see Setting Up a Spectre Analysis, or refer to your simulator manual.


© 2023 Cadence Design Systems, Inc.

# Specifying Order for Analyses

When you set up an analysis using the *Choosing Analyses* form, the analysis is added at the end of the list of analyses displayed in the Setup assistant of the ADE Explorer window.




During simulation, the analyses are run in the same order as they appear in this list.

However, for some simulators the analyses need to be run in a pre-defined sequence. For example, the `pss` analysis should run before `pac`. Therefore, every time you change the position of an analysis, the tool checks whether all the analyses are placed in the correct order as required by the selected simulator. If you place an analysis at an incorrect position, it moves back to its original position and an appropriate message is displayed.

Perform the following steps to change the order of the analyses and specify the order in which you want the analyses to run:

- To change the position of a single analysis, drag and drop the analysis to the desired position in the list.
- To change the position of multiple analyses together, press the *Control* key, click the analyses which you want to reorder to highlight them, and drag them to the desired position in the list.


© 2023 Cadence Design Systems, Inc.

# Setting Up a Spectre Analysis

Perform the following steps to set up analyses for Spectre:

1. Choose *Analyses – Choose*.

   The *Choosing Analyses* form appears.

2. In the *Analysis* field, select an analysis.

   The *Choosing Analyses* form shows the parameters of the selected analysis.

3. Set the options and select the *Enabled* check box.
4. Click *Apply* to save the settings.
5. Choose another analysis and perform the same steps to complete the setup.

For help on setting up a particular analysis, refer to *Analysis Statements* chapter in the *Spectre Circuit Simulator Reference*.

You can run the following analyses for Spectre:

- Transient Analysis on page 250
- Transient Noise Analysis on page 269
- DC Analysis on page 278
- AC Analysis on page 285
- Noise Analysis on page 288
- S-parameter Analysis on page 292
- Transfer Function Analysis on page 303
- Sensitivity Analysis on page 306
- DC Match Analysis on page 309
- AC Match Analysis on page 314
- Stability Analysis on page 323
- Pole-Zero Analysis on page 329
- Other Spectre Analyses on page 340


© 2023 Cadence Design Systems, Inc.

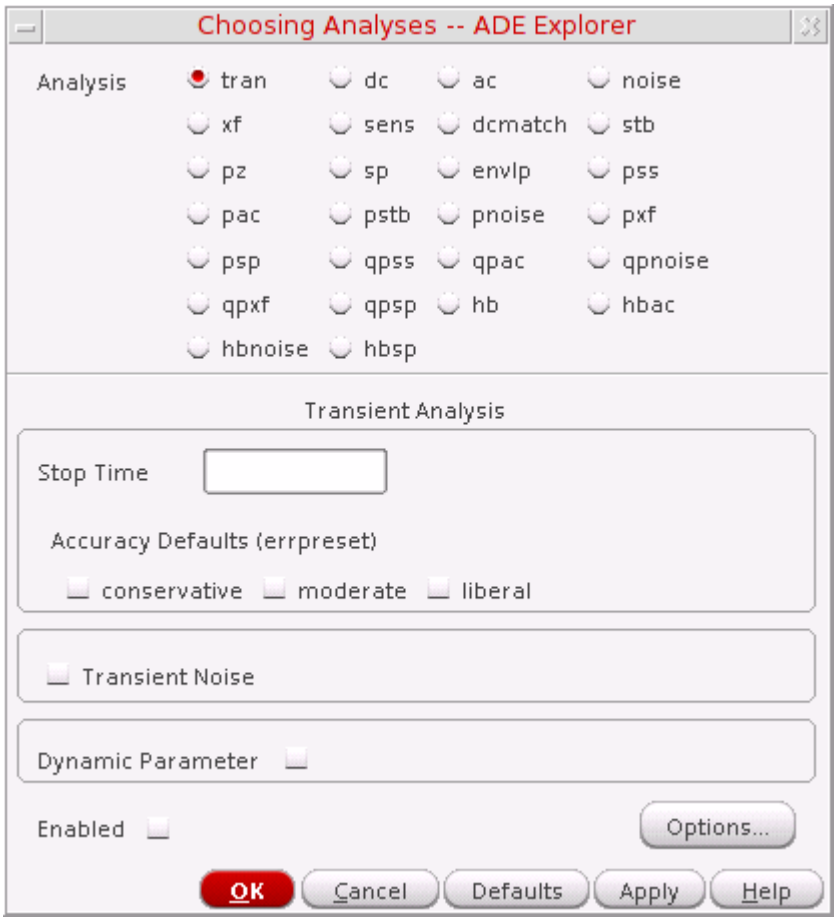
## Transient Analysis

Transient analysis computes the transient response of a circuit over an interval. The initial condition is taken to be the DC steady-state solution.

Perform the following steps to set up a transient analysis:

1. In the *Choosing Analyses* form, choose *tran*.

   The *Choosing Analyses* form is updated to show the additional fields for transient analysis.




2. Enter the stop time.
3. Select the default accuracy level for the simulation.

   For more information, see the documentation for the `errpreset` parameter in *Spectre Circuit Simulator Reference*.


© 2023 Cadence Design Systems, Inc.

4. If you want to perform transient noise analysis, select the *Transient Noise* check box. For more information, see Transient Noise Analysis.

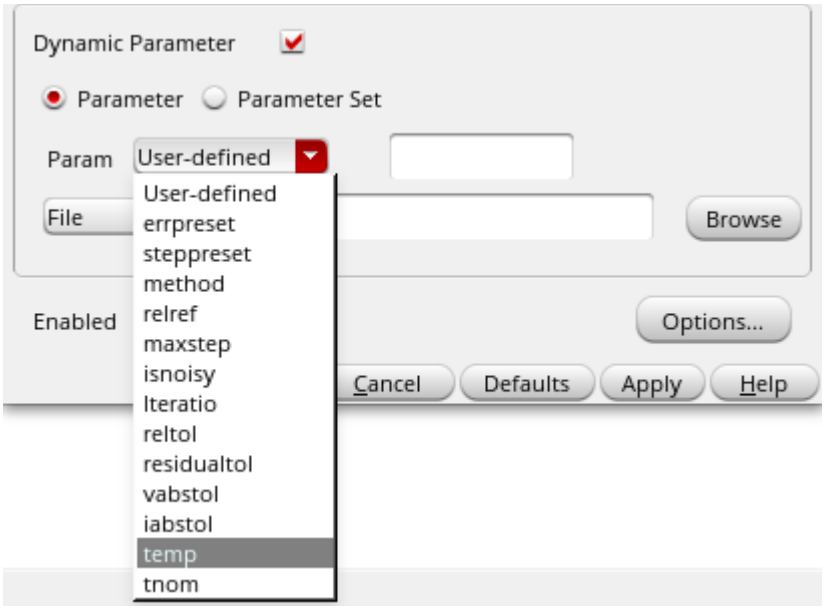
5. Select the *Dynamic Parameter* check box to vary temperature, design parameters, options, or transient analysis parameters (such as `reltol`, `residualtol`, `vabstol`, `iabstol`, `isnoisy`) during transient simulation. Then, perform the following steps:

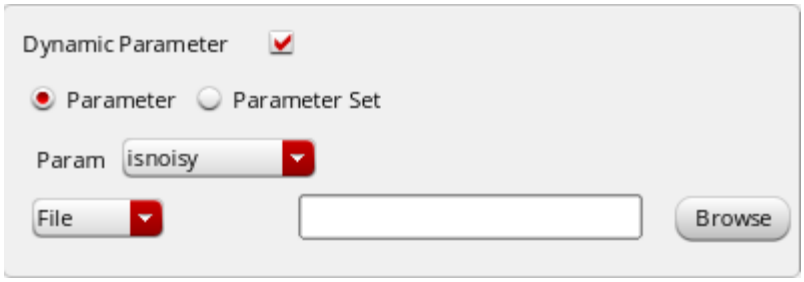
   a. Specify the parameter name in the test box for the *Param* option.

   b. Perform one of the following steps:

      - From the drop-down list, select *File* and specify the path to the file that contains the parameter values that need to be varied with time.

        Set the environment variable `maestro.gui donotExpandNameForparam_file` to specify whether or not the specified file path should be resolved to its absolute path.




The format of the file can be as shown below:


© 2023 Cadence Design Systems, Inc.

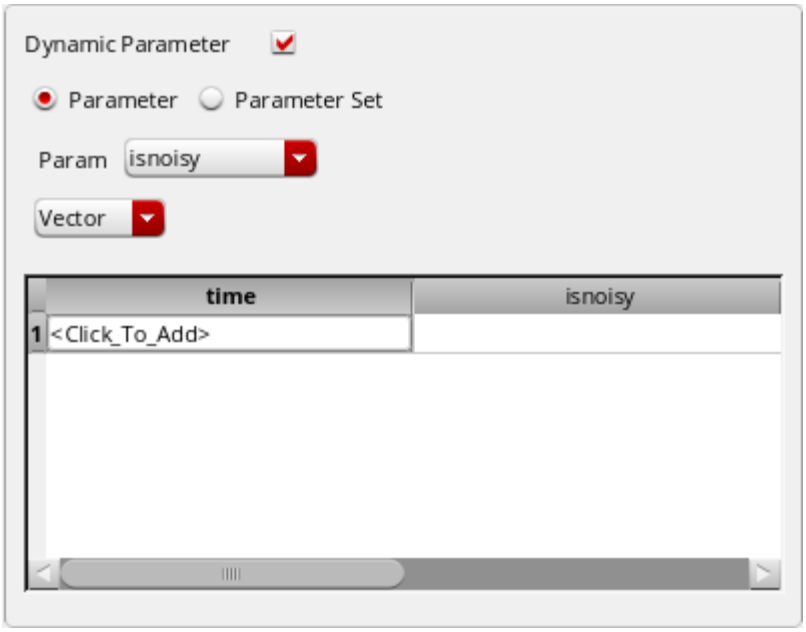
```
; comments
tscale tscale_value
time  value
20  50.0
30  60.0
```

Your comment line starts with ; at the beginning of a line. `tscale` is keyword and `tscale_value` is a value such as 1.0e-6, 1.0e-9, and so on, which is applied to each time point under the time column. `time` and `value` are two key words which are used to identify the time and value columns. The values under the time column define the time points and each time point is scaled by `tscale_value`. The values under the value column define the values for the dynamic parameter.

No unit is supported in the file format.

❍ From the drop-down list, select *Parameter vector*, then specify the time and value in the *Time* and *Value* fields, and click the *Add* button.

The time value pairs are added to the table.

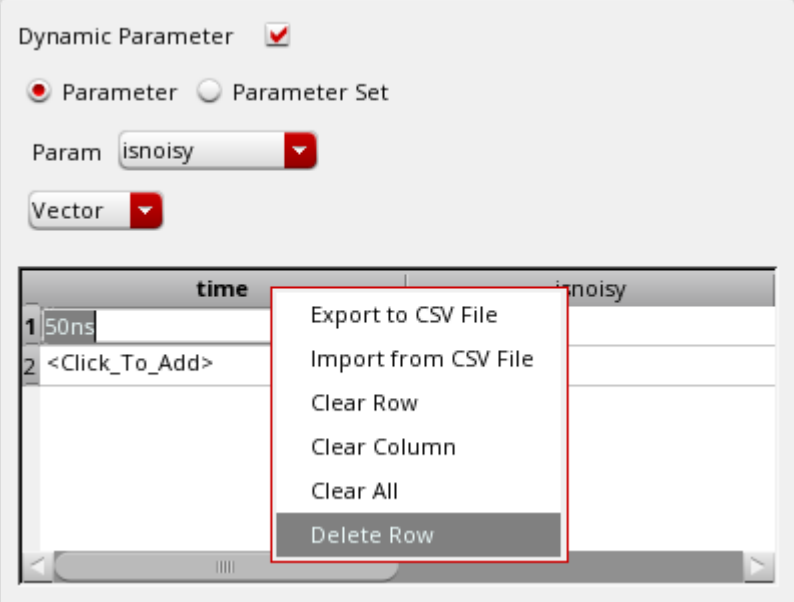



To modify an existing time value pair, select the row (double-click), and update the values in the *time* and *value* columns.

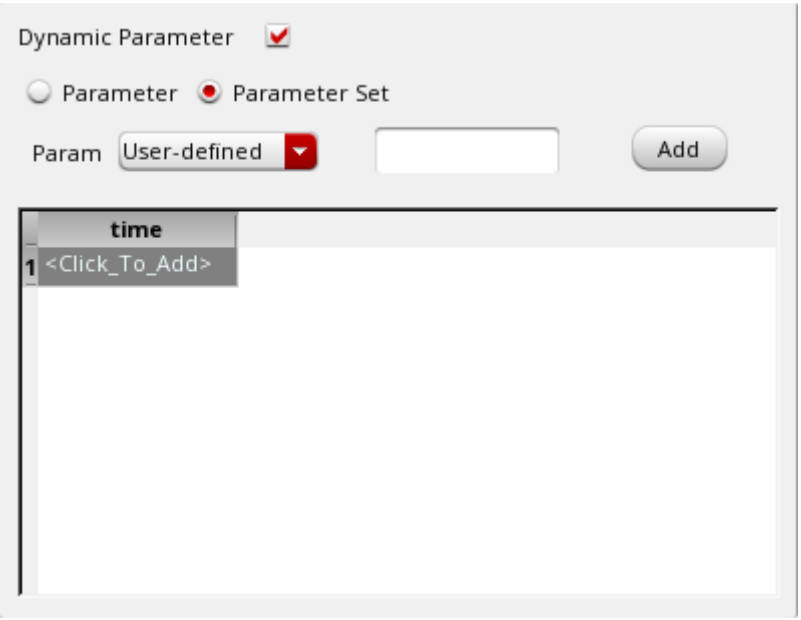
To add more time value pairs after modifying an existing row, and click Enter, and specify the time value pairs.


© 2023 Cadence Design Systems, Inc.

To delete a time value pair, select the row in the table and right-click to choose *Delete Row*.



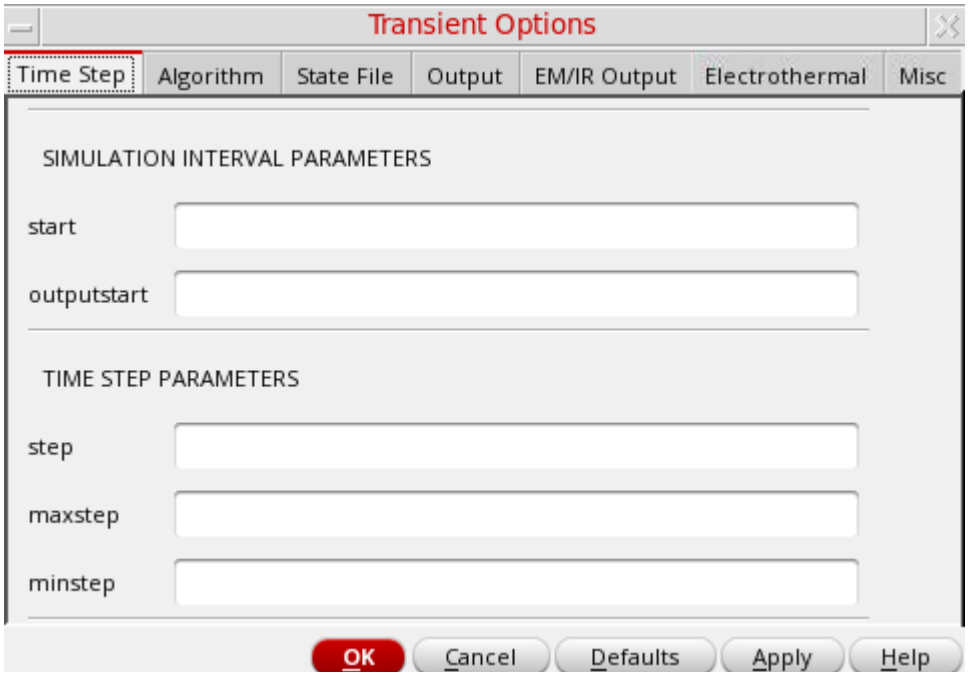

**c.** You can select a set of parameters by clicking *Parameter Set*, selecting the *Param* and follow the same steps as explained above.





© 2023 Cadence Design Systems, Inc.

6. Click *Options* to display the *Transient Options* form.




### Transient Options Form

The Transient Options form lets you specify the settings for a transient analysis that computes the transient response of a circuit over an interval. You can adjust these settings in several ways to meet the needs of your simulation. Setting the options that control the error tolerances, the integration method, and the amount of data saved lets you choose between maximum speed and greatest accuracy in a simulation.The form contains the following tabs.

| Tab | Description |
|---|---|
| Time Step | Lets you specify the values for simulation interval and time step parameters. |
| Algorithm | Lets you specify the values for initial condition, convergence, integration method, accuracy, and newton parameters. |
| State File | Lets you specify the values for state file and save-restart parameters. |
| Output | Lets you specify the values for output parameters. |
| EM/IR Output | Lets you specify the values for EMIR parameters. |
| Fault | Lets you specify the values for fault simulation transient options. |
| Electrothermal | Lets you specify the options for electrothermal analysis. |


© 2023 Cadence Design Systems, Inc.

| Tab | Description |
|---|---|
| Misc | Lets you specify the miscellaneous options for transient analysis. |

## Time Step

The following table describes the fields available on the *Time Step* tab of the Transient Options form.

| Field | Description |
|---|---|
| ***SIMULATION INTERVAL PARAMETERS*** | Specifies the values for simulation interval parameters. |
| *start* | Specifies the start time for the simulation run. |
| *outputstart* | Specifies the time after which the outputs are to be saved. |
| ***TIME STEP PARAMETERS*** | Specifies the values for time step parameters. |
| *step* | Specifies the minimum time step used by the simulator solely to maintain the aesthetics of the computed waveforms. |
| *maxstep* | Specifies the maximum time step. The default is derived from `errpreset`. |
| *minstep* | Specifies the minimum time step.<br>If specified, the error tolerance requirements may be ignored when step size is less than minstep. |

## Algorithm

The following table describes the fields available on the *Algorithm* tab of the Transient Options form.

| Field | Description |
|---|---|
| ***INITIAL CONDITION PARAMETERS*** | Specifies the values for initial condition parameters. |
| *ic* | The value to be used to set initial condition. Possible values are *dc*, *node*, *dev*, and *all*. |


© 2023 Cadence Design Systems, Inc.

| Field | Description |
|---|---|
| *skipdc* | Specifies whether DC analysis should be skipped or not. Possible values are no, yes, useprevic, waveless, rampup, autodc, and, sigrampup.<br>■ *no*: Initial solution is calculated using normal DC analysis (default).<br>■ *yes*: Initial solution is given in the file specified by the `readic` parameter or the values specified on the `ic` statements.<br>■ *useprevic*: Uses the converged initial condition from previous analysis as `ic` or `ns`.<br>■ *waveless*: Same initial solution as `skipdc=yes`, but the waveform production in the time-varying independent sources is disabled during the transient analysis. Independent source values are fixed to their initial values (not their DC values).<br>■ *rampup*: Independent source values start at `0` and ramp up to their initial values in from start to `rampuptime`. If rampuptime is not given, `rampuptime` will be set to `rampupratio*stop`. After that their values remain constant. Zero initial solution is used.<br>■ *autodc*: Same as `skipdc=waveless` if a nonzero initial condition is specified. Otherwise, same as `skipdc=rampup`.<br>■ *sigrampup*: Independent source values start at `0` and ramp up to their initial values in the first phase of the simulation. Unlike `skipdc=rampup`, the waveform production in the time-varying independent source is enabled after the `rampup` phase. The `rampup` simulation is from the start parameter. If the start parameter is not specified, the default start time is set to `rampupratio*stop`. |
| *readic* | Specifies the file that contains the initial conditions. |
| ***INITIAL CONDITION PARAMETERS FOR OSCILLATOR*** | Specifies the values for initial condition parameters for oscillators. |


© 2023 Cadence Design Systems, Inc.

| Field | Description |
|---|---|
| *calculate ic automatically* | Enables linear IC method to calculate initial conditions automatically from a type of stability analysis in the range [0.5*oscfreq, 1.5*oscfreq]. Overrides the user-defined initial conditions if instability is detected. The default is `no`.<br><br>Possible values are `no` and `yes`. |
| *Estimated frequency* | Specifies the estimation of the oscillation frequency when the linear IC method is enabled. The default is `0.0`. |
| ***CONVERGENCE PARAMETERS*** | Specifies the values for convergence parameters. |
| *readns* | Specifies a file that contains an estimate of the initial transient solution. |
| *cmin* | Specifies the minimum capacitance from each node to ground. |
| ***INTEGRATION METHOD PARAMETERS*** | Specifies the values for integration method parameters. |


© 2023 Cadence Design Systems, Inc.

| Field | Description |
|---|---|
| *method* | Specifies the integration method. The default derived from errpreset.<br>■ *euler*: Backward-Euler is used exclusively.<br>■ *trap*: An advanced version of trap that uses all three integration methods.<br>■ *traponly*: Trapezoidal rule is used almost exclusively.<br>■ *gear2*: Backward-Euler and second-order Gear are used.<br>■ *gear2only*: Gear's second-order backward-difference method is used almost exclusively.<br>■ *gear3*: Backward-Euler and third-order Gear are used.<br>■ *gear3only*: Gear's third-order backward-difference method is used almost exclusively.<br>■ *gear2gear3*: Backward-Euler is used along with second-order and third-order backward-difference method is used almost exclusively.<br>■ *trapgear2*: Backward-Euler and the trapezoidal rule are used with Gear's second-order backward-difference method.<br>■ *trapeuler*: Backward-Euler and the trapezoidal rule are used. |
| ***ACCURACY PARAMETERS*** | Specifies the values for accuracy parameters. |
| *relref* | Specifies the reference used for the relative convergence criteria. Default is derived from `errpreset`.<br>■ *pointlocal*: Compares the relative errors in quantities at each node to that node alone.<br>■ *alllocal*: Compares the relative errors at each node to the largest values found for that node alone for all past time.<br>■ *sigglobal*: Compares relative errors in each circuit signal to the maximum for all signals at any previous point in time.<br>■ *allglobal*: Same as `relref=sigglobal`, except that it also compares the residues (KCL error) for each node to the maximum of that node's past history |


© 2023 Cadence Design Systems, Inc.

| Field | Description |
|---|---|
| *Iteratio* | Specifies the ratio used to compute LTE tolerances from Newton tolerance. The default is derived from `errpreset`. |
| ***NEWTON PARAMETERS*** | Specifies the values for newton parameters. |
| *maxIters* | Specifies the maximum number of Newton iterations per transient integration time step |
| *restart* | Restarts the DC solution from scratch if any condition has changed. If not, use the previous solution as initial guess.<br>Possible values are *yes* and *no*. |

## State File

The following table describes the fields available on the *State File* tab of the Transient Options form.

| Field | Description |
|---|---|
| ***STATE FILE PARAMETERS*** | This section lets you specify state file parameters. |
| *write* | Specifies the file to which initial transient solution is to be written. |
| *writefinal* | Specifies the file to which final transient solution is to be written. |
| ***SAVE-RESTART PARAMETERS*** | Specifies the values for save-restart parameters. |
| *saveclock* | Saves the transient analysis data periodically on the wall clock time. The default is 1800s for Spectre.<br>This parameter is disabled in Spectre APS mode by default. |
| *saveperiod* | Saves the transient analysis data periodically on the simulation time. |
| *savetime* | Saves the analysis states into files at the specified time points. |
| *savefile* | Saves the analysis states into the specified file. |
| *recover* | Specifies the file to be restored. |


© 2023 Cadence Design Systems, Inc.

## Output

The following table describes the fields available on the *Output* tab of the Transient Options form.

| Field | Description |
|---|---|
| ***OUTPUT PARAMETERS*** | Specifies the values for output parameters. |
| *save* | Specifies the signals to output.<br>■ *selected* - voltage waveforms for nodes saved by the user are created, if no node voltage is saved, then *allpub* is used.<br>■ *allpub* - all node voltages, voltage source currents, inductor currents, and iprobe currents are saved.<br>■ *lvlpub* - all node voltages, voltage source currents, inductor currents, and iprobe currents down nestlvl levels are saved.<br>■ *nooutput* - disables any waveform writing.<br>■ *all* - same as *allpub*, additionally saves internal nodes of active devices.<br>■ *lvl* - same as *lvlpub*, additionally saves internal nodes of active devices. |
| *nestlvl* | Specifies the levels of subcircuits to output. |
| *compression* | Perform global waveform compression. Possible values are `no`, `all`, `wildcardonly` and `yes`. |
| *complvl* | Enables waveform compression for specified hierarchy level and below (top level=1). All levels above specified level are not compressed. Complvl has higher priority than global compression statement. |
| *compreltol* | Specifies the relative tolerance for waveform compression. |
| *compvabstol* | Specifies the absolute voltage tolerance for waveform compression. |
| *compiabstol* | Specifies the absolute current tolerance for waveform compression. |
| *flushpoints* | Flushes all unwritten data periodically from the buffer to the outputs after calculating the specified number of points |


© 2023 Cadence Design Systems, Inc.

| Field | Description |
|---|---|
| *flushtime* | Flushes unwritten data periodically from the buffer to the outputs after the specified time has elapsed. |
| *flushofftime* | Specifies the Real time to stop flushing outputs. |
| ***STROBING PARAMETERS*** | Specifies the values for strobing parameters. |
| *oppoint* | Determines whether the operating point information should be computed for initial time step; if yes, where should it be printed (screen or file). Possible values are `no`, `screen`, `logfile`, and `rawfile`. |
| *skipstart* | Specifies the time to start skipping output data. |
| *skipstop* | Specifies the time to stop skipping output data. |
| *skipcount* | Saves only one of every `skipcount` points. |
| *strobeperiod* | Specifies the output strobe interval (in seconds) of transient time. |
| *strobeoutput* | Specifies which time points to output during strobe. Possible values are `strobeonly`, `all`, and `none`. |
| *strobedelay* | Specifies the delay (phase shift) between the `skipstart` time and the first strobe point. |
| ***NEWTON PARAMETERS*** | Specifies the values for Newton parameters. |


© 2023 Cadence Design Systems, Inc.

| Field | Description |
|---|---|
| *Save Final Op Pt* | Generates the info statements for final operating point into the control file and related data into in psf file. If you do not want the results to be saved, select No. When this option is disabled, no final operating point data is generated.<br>An example of a netlist:<br>`simulatorOptions options reltol=1e-3 vabstol=1e-6 iabstol=1e-12 temp=27 \`<br>`  tnom=27 scalem=1.0 scale=1.0 gmin=1e-12 rforce=1 maxnotes=5 maxwarns`<br>`  digits=5 cols=80 pivrel=1e-3 sensfile="../psf/sens.output" \`<br>`  dochecklimit=no checklimitdest=psf inlinesubcktcurrent=subkt \`<br>`  colonasdelimiter=none`<br>`tran tran stop=30n errpreset=moderate annotate=status save=selected \`<br>`  oppoint=no maxiters=5`<br>`finalTimeOP info what=oppoint where=rawfile`<br>`dcOp dc write="spectre.dc" maxiters=150 maxsteps=10000 annotate=status`<br>`dcOpInfo info what=oppoint where=rawfile`<br>`stb stb start=1 stop=1G dec=10 probe=VFB annotate=status`<br>`save OUT`<br>`saveOptions options save=allpub pwr=none currents=selected useprobes=no \`<br>`  saveahdlvars=selected`<br>These operating points are saved in the *tranOp* directory in the results database and can be plotted in the graph window.<br>Environment variable: finalTimeOp |
| *Final Op Other Params* | Outputs all node voltages into raw data. |
| ***INFOTIMES SETTINGS*** | Specifies the infotimes settings. |
| *infotimes* | Specifies the times when the analysis specified by infoname is performed. |
| *infoname* | Names of info analyses to be performed at the time point in the `infotimes` array. |
| *Tran Info other Params* | Lets you set up additional parameters for operating points and infotimes in Transient analysis. |
| ***ACTIMES SETTINGS*** | Specifies the actimes settings. |
| *Enabled* | Enables the fields *actimes* and *acnames* and indicates that the arrays specified through them must be netlisted. |


© 2023 Cadence Design Systems, Inc.

| Field | Description |
|---|---|
| *actimes* | Specifies the time points when the analyses specified in `acname` array are performed.<br><br>You can also specify variables in the actimes field using the VAR() syntax as shown below:<br><br>`VAR("myVariable")`<br><br>From Spectre 17.10 ISR 4, If variable `actime_pair` is set as `yes`, the `actimes` and `acnames` are used as pairs, which means, apply acname1 to actime1, acname2 to actime2, and so on. If `actime_pair` is set as `no`, only one `acname` to `actime` is allowed. It is the same for fields *infonames* and *infotimes*.<br><br>**Note:** Virtuoso ADE allows only one infoname or acname. To successfully generate a netlist, it sets `actime_pair=no` and `infotime_pair=no`. |
| *acnames* | Names of ac, noise, sp, stb, or xf analyses to be performed at each time point in the `actimes` array. The named small-signal analyses are not run separately, but only as part of the transient analysis. |

## EM/IR Output

The following table describes the fields available on the *EM/IR Output* tab of the Transient Options form.

**Note:** The *EM/IR Output* tab is not available when the simulation performance mode is set to Spectre FX.

| Field | Description |
|---|---|
| ***EM/IR PARAMETERS*** | This section lets you specify the values for the parameters available for EMIR analysis. |
| *emirformat* | Specifies the format of the EM/IR database file.<br><br>Possible values are `none` and `vavo`. |
| *emirstart* | Specifies the EM/IR start time. |
| *emirstop* | Specifies the EM/IR stop time. |


© 2023 Cadence Design Systems, Inc.

| Field | Description |
|---|---|
| *emirfile* | Specifies the Name of the EM/IR database file. Default is `%A_emir_vavo.db`. The file will be output to raw directory |

## Fault

The following table describes the fields available on the *Fault* tab of the Transient Options form.

**Note:** The *Fault* tab is not available when the simulation performance mode is set to Spectre FX.

| Field | Description |
|---|---|
| ***FAULTSIMULATION TRANSIENT OPTIONS*** | This section lets you specify the values for fault simulation transient options. |
| *Fault Points Method* | Specifies the method to be used for transient fault analysis.<br>■ Time *Points List*: Displays and enables the Fault Time Points field in which you can specify a space-separated list of time points at which the transient fault analysis is to be performed.<br>These fault time points specified in the run options form for fault simulation are used for all tests in the setup. However, if a particular test contains specific time points in the transient analysis options, the test-specific values take precedence over the global values in the run options.<br>■ *Start/Step/Stop*: Displays and enables the fields to set the start, step and stop time for transient fault analysis.<br>❑ Start – The first test point or start time<br>❑ Step – The time interval between test points<br>❑ Stop – The last test point or stop time |

## Electrothermal

The following table describes the fields available on the *Electrothermal* tab of the Transient Options form.


© 2023 Cadence Design Systems, Inc.

**Note:** This tab is not available when the simulation performance mode is set to Spectre FX.

| Field | Description |
|---|---|
| *Enable Electrothermal Analysis* | Enables or disables thermal analysis.<br>Default value: `unselected` |
| *Thermal Configuration File* | Specifies the location of the thermal configuration file.<br>Default value: `""` |
| *Thermal Analysis Method* | Select the thermal analysis mode.<br>Possible Values: `Steady State` and `Dynamic`<br>Default value: `Steady State` |
| *Trise Limit* | Specifies the upper limit to be used for updating the device temperatures. This helps to prevent device temperatures from going out of range during electrical simulation. Unit is degree centigrade.<br>Default value: `300` |
| *Report Updated Parasitic Resistor Temperatures* | Specifies whether or not to report the resistor temperature values in thermal reports.<br>Default value: `Yes` |
| *Number of Electrothermal Iterations* | Specifies the number of iterations to run when using the steady state thermal analysis method.<br>Default value: `2` |
| *Sorting Method* | Device sorting order in debug reports (`.dbg_trise_iter*` and `.thermal_pwr.iter*`) for steady state thermal analysis.<br>Possible values:<br>■ `Temperature Rise (trise values)`<br>■ `Power`<br>Default value: `Temperature Rise` |
| *Max Devices in Report* | Specifies the number of instances to be included in steady state thermal analysis reports. |


© 2023 Cadence Design Systems, Inc.

| Field | Description |
|---|---|
| *Chip Bounding Box* | Specifies a vector of four numbers [lower-left-x-coordinate lower-left-y-coordinate upper-right-x upper-right-y] representing a user-defined chip bounding box. By default, the chip bounding box is calculated from the device geometries so that a tight bounding box containing all the devices is used.<br><br>Default value: `nil` |
| *Devices in Thermal Report* | Specifies a list of instance names for which temperature and power are to be included in dynamic thermal analysis reports.<br><br>Default value: `nil` |
| *Thermal Time Step* | Specifies the thermal time step (in seconds) for dynamic thermal solver. Set this parameter to zero to get identical electrical and thermal time steps.<br><br>Default value: `1.0` |
| *Trise for Thermal Update* | Specifies the Trise threshold for device temperature update.<br><br>Default value: `0.0` |
| *Save Channel Layer* | Specifies whether the Trise data is to be saved for the channel layer only or for all layers.<br><br>When set to `Yes`, the data is saved only for the channel layer. Otherwise, the data is saved for all layers.<br><br>Default value: `No` |
| *Save Layer* | Specifies a list of layer names for which trise will be reported in the thermal `.temp_grid` file.<br><br>Default value: `""`<br><br>Use this field when you want to save the data for a selected set of layers. To save data for all the layers, select *No* from the *Save Channel Layer* drop-down list. |
| *Additional Options* | Placeholder where you can specify any additional option for Spectre electrothermal analysis. |


© 2023 Cadence Design Systems, Inc.

## Misc

The following table describes the fields available on the *Misc* tab of the Transient Options form.

| Field | Description |
|---|---|
| ***Annotation PARAMETERS*** | Specifies the values for annotation parameters. |
| *annotate* | Specifies the degree of annotation.<br>Possible values are `no`, `title`, `sweep`, `status`, and `steps`. |
| *annotatedigits* | Specifies the number of significant digits to be annotated to the transient time in the simulation log file. Possible values are `0`, `1`, `2` ...`16`. If set to 0, the number of significant digits is dynamic as per the current time and step. If set to n (where n = 1, 2 ...16), the number is equal to n. |
| ***CAPTAB PARAMETERS*** | Specifies the values for captab parameters. |
| *captab* | Prints node-to-node capacitance. |
| *timed* | Specifies the time points at which the check needs to be performed.<br>This check box is displayed only when *captab* is selected. |
| *threshold* | Specifies the threshold value for printing capacitances (ignore capacitances smaller than this value).<br>This field is displayed only when *captab* is selected. |
| *detail* | Specifies how detailed should the capacitance table be.<br>Possible values are `node`, `nodetoground`, and `nodetonode`.<br>This check box is displayed only when *captab* is selected. |
| *sort* | Specifies how to sort the capacitance table.<br>Possible values are `name` and `value`.<br>This check box is displayed only when *captab* is selected. |


© 2023 Cadence Design Systems, Inc.

| Field | Description |
|---|---|
| *intrinsic_cap_merge* | Merges the internal captab node with the external node.<br>Possible values are `yes` and `no`.<br>The default value is `no`.<br>You can also specify this parameter in the *Name* column of the *Save circuit information analysis* group box in the Save Options form. This is possible only when the *What* column is set to *captab*.<br>This check box is displayed only when *captab* is selected. |
| ***THERMAL NODE PARAMETERS*** | Specifies the values for thermal node parameters. |
| *enable thermal node* | Lets you specify thermal nodes. Thermal nodes provide you the ability to dynamically analyze the effect of temperature on the electrical behavior of the design during simulation |
| *Trise delta* | Specifies the temperature rise delta value for the thermal nodes. |
| *thermal model fille* | Lets you specify a thermal model file.<br>This field is displayed only when *enable thermal node* is selected. |
| *Instance* | Lets you select an instance from the schematic.<br>This column is displayed only when *enable thermal node* is selected. |
| *Thermal Pin Name* | Specifies the thermal pin name to be added to the table.<br>This column is displayed only when *enable thermal node* is selected. |
| *Add* | Adds the specified instance and thermal pin name as an entry in the table. |
| *Delete* | Deleted the selected entry from the table. |
| *Export* | Lets you export the setup from the table to a CSV file. |
| *Load* | Lets you import the setup from a CSV file. |
| ***ADDITIONAL PARAMETERS*** | Specifies the additional parameters required for Transient Analysis. |


© 2023 Cadence Design Systems, Inc.

| Field | Description |
|---|---|
| *additionalParams* | Lets you specify any additional parameters required for Transient analysis. |

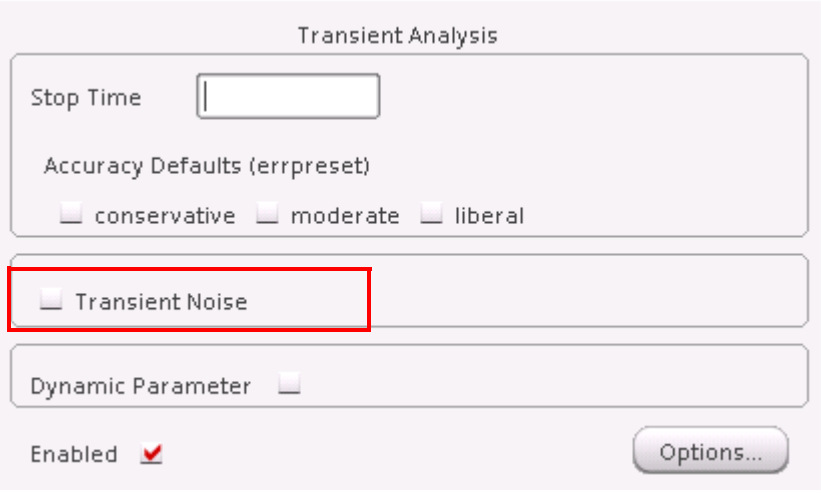
## Transient Noise Analysis

The current transient analysis has been extended to support transient noise analysis. Transient noise provides the benefit of examining the effects of large signal noise on many types of systems. It gives you an opportunity to examine the impact of noise in the time domain on various circuit types, without requiring access to the SpectreRF analyses. This capability is accompanied by enhancements to several calculator functions, allowing you to calculate multiple occurrences of measurements, such as risetime and overshoot.

Spectre provides both a single run and multiple run method of simulating transient noise. The single run method, which involves a single transient run over several cycles of operation, is best suited for applications where undesirable start-up behavior is present. The multiple run method, which involves a statistical sweep of several iterations over a single period, is recommended for users who are able to take advantage of distributed processing.

Perform the following steps after setting up transient analysis:

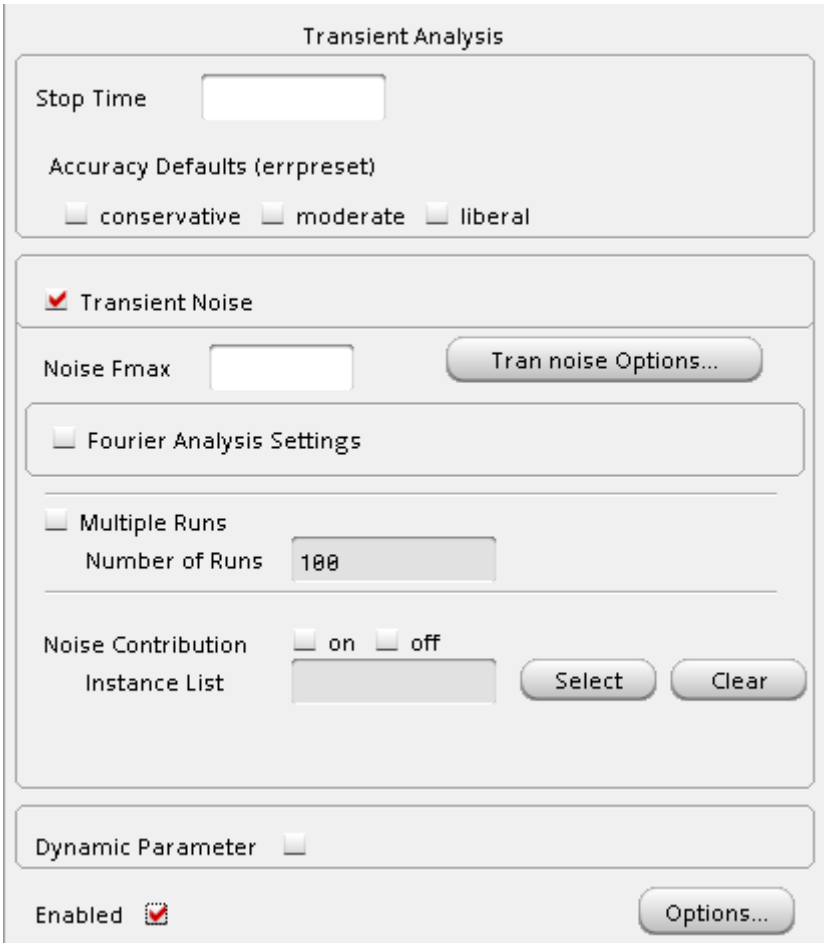
1. In the main *Transient Analysis* (*Analyses – Choose – tran)* form, select the *Transient Noise* check box to enable this feature.




For information on how to set up to run an analysis in ADE, see Setting Up a Spectre Analysis.


© 2023 Cadence Design Systems, Inc.

2. When the *Transient Noise* check box is selected, the *Choosing Analyses* form is updated to show the additional fields for transient noise analysis.

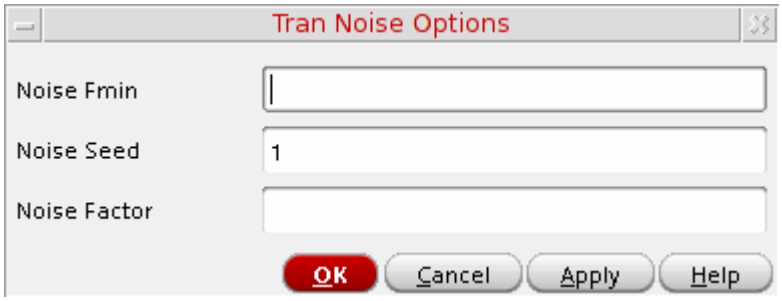



3. Set the following parameters to calculate noise during a transient analysis.

   a. *Noise Fmax*—The bandwidth of pseudorandom noise sources. A valid (nonzero) value turns the noise sources on during transient analysis. The maximal time step of the transient analysis is limited to `1/Noise Fmax`.


© 2023 Cadence Design Systems, Inc.

**b.** Click the *Tran noise Options* button to open the Tran Noise Options form, as shown below, and set the additional parameters required to calculate noise during transient analysis.




**c.** *Noise Fmin*—If specified, the power spectral density of the noise sources depends on the frequency in the interval from Noise Fmin to Noise Fmax. Below Noise Fmin, the noise power density is constant. If Noise Fmin=Noise Fmax, then only white noise would be included, and noise sources are evaluated only at Noise Fmax for all models. `1`/Noise Fmin should be smaller than the requested time duration of transient analysis. The default value is `1hz`.

**d.** *Noise Seed*—Seed for the random number generator (used by the simulator to vary the noise sources internally). Specifying the same seed allows you to reproduce a previous experiment. The default value is `1`.

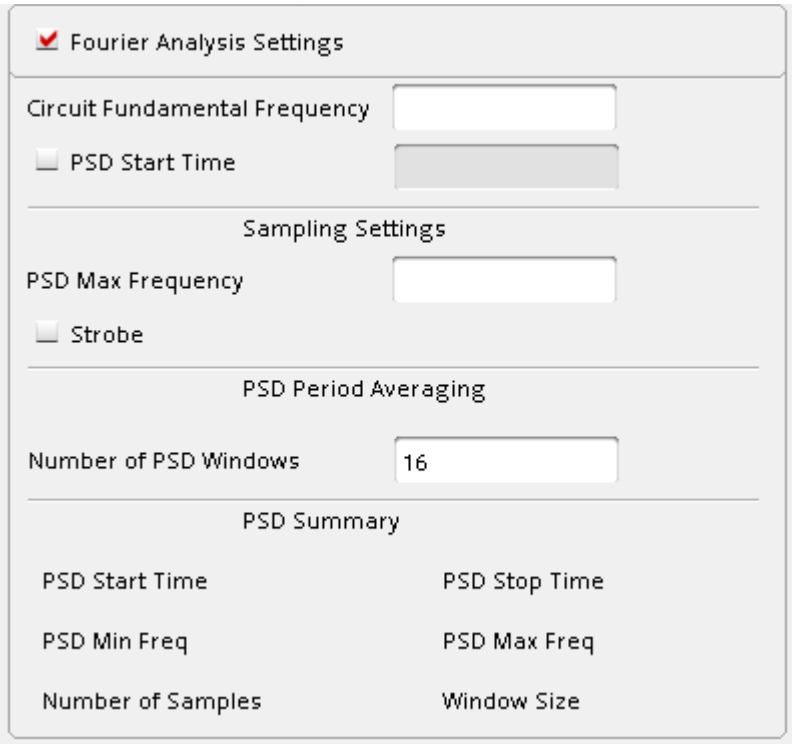
**e.** *Noise Factor*—Noise factor applied to all the generated noise. It can be used to artificially inflate small noise to make it visible over the transient analysis numerical noise floor, but it should be small enough to maintain the nonlinear operation of the circuit.

**Note:** If any of these parameters is not specified, it will not be netlisted.This results in the simulator using its internal default values.

**f.** *Fourier Analysis Settings*—Select this check box if you want to perform Fourier analysis. The additional options required for Fourier analysis appear on the form.


© 2023 Cadence Design Systems, Inc.

See Fourier Analysis Settings for more information.



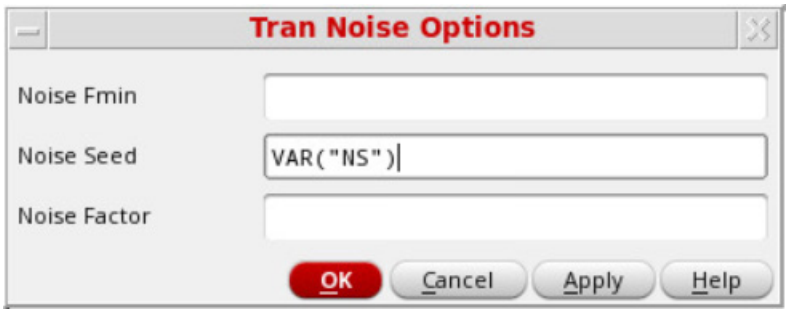

**g.** *Multiple Runs*—Select this check box to perform multiple transient noise analysis runs and enter the number of times the transient-noise analysis has to be run in the *Number of Runs* field. The default for this option is *100* (number of runs).

Spectre provides both a single run and multiple run methods of simulating transient noise. The single run method, which involves a single transient run over several cycles of operation, is best suited for applications where undesirable start-up behavior is present. The multiple run method, which involves a statistical sweep of several iterations over a single period, is recommended for users who are able to take advantage of distributed processing. If enabled, it will perform multiple runs in series in a single Spectre simulation.

To perform multiple runs in parallel, disable the *Multiple Runs* check box and click the *Tran noise Options* button. The *Tran noise Options* form is displayed. In this


© 2023 Cadence Design Systems, Inc.

form, specify a variable in the *Noise Seed* field. For example, let us specify a variable `Var("NS")`.




You can now sweep this variable and use a job policy to run the sweeps in parallel.

*Tip*

If you switch from single run noise analysis to multi run analysis, adjust the *Stop Time* appropriately. For example, specify 5 iterations of 20us for a single run of `100us`.

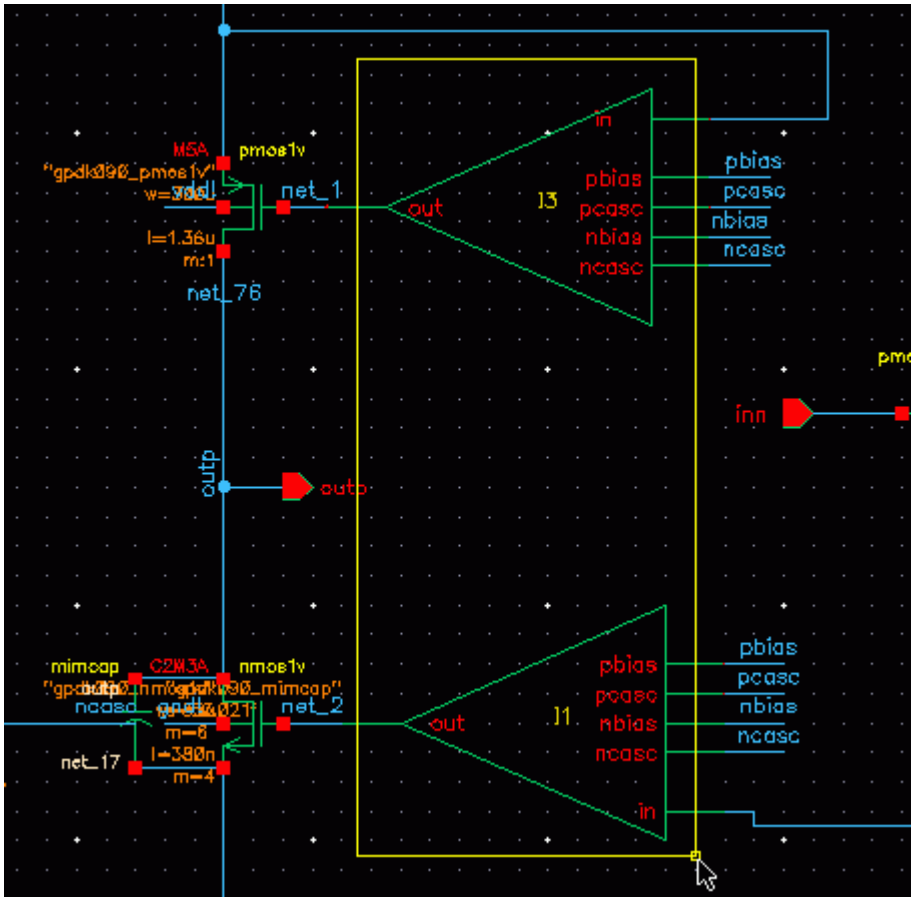
**h.** *Noise Contribution*—You can use this field to narrow down the main source of circuit noise by specifying the list of devices and subcircuit instances that are noisy or noise free by performing the following steps:

- Select one of the following check boxes:
  - *on*—To specify the list of instances to be considered as noisy during transient noise analysis.
  - *off*—To specify the list of instances to be considered as not noisy during transient noise analysis.
- In the *Instance List* field, enter the instance names of the devices and subcircuit instances separated by spaces.

  To select instances from the schematic, do the following:
  - Click the *Select* button next to the *Instance List* field to open the schematic.
  - Select one or more instances on the schematic. To select more than one instance at a time, hold down the *Shift* key and click instances or 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


© 2023 Cadence Design Systems, Inc.

following example, instances `I1` and `I3` that are within the yellow bounding box are included in the selection.




- Press the *Esc* key when you are done. The selected instances are displayed in the *Instance List* field.

For more information about the transient noise parameters, refer to the section *Calculating Transient Noise* in the *Spectre Classic Simulator, Spectre APS, Spectre X, and Spectre XPS User Guide.*

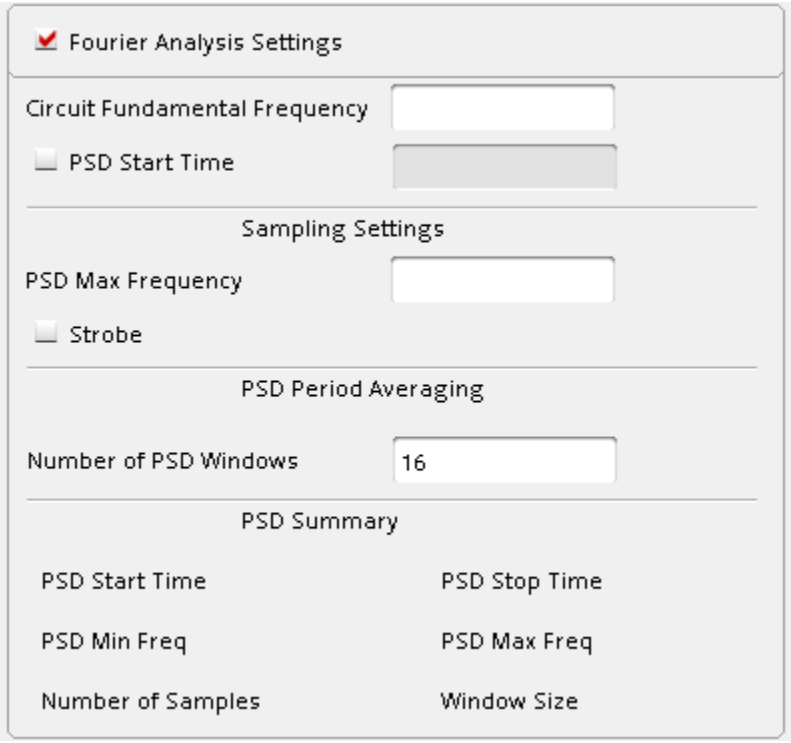
You can specify the options corresponding to transient noise analysis in the *Transient Options* form.

### *Fourier Analysis Settings*

Fourier analysis is used to calculate the Power Spectral Density (PSD) waveforms at the output nodes/probes. PSD is important to predict the noise behavior, especially for non-


© 2023 Cadence Design Systems, Inc.

periodic circuits. The options described below, enable easier computation of Fourier spectrum measurements after transient noise:




- *Circuit Fundamental Frequency*—Specifies the fundamental frequency of the circuit at the output node. Providing a fundamental frequency reduces the number of numerical errors generated during Fourier Transform and decreases the noise in PSD waveforms.
- *PSD Start Time*—Specifies the start time for PSD analysis. It allows for an initial time to let transient behaviors and time steps to stabilize before applying Fourier analysis. The default value is `1`/(Circuit Fundamental Frequency).
- *PSD Max Frequency*—Specifies the maximum frequency required at the output of the PSD waveforms.
- *Strobe*—Helps in decreasing errors caused due to interpolation in Fourier analysis.
- *Number of PSD Windows*—The number of windows (frames) that will be used to average the results of Fourier transform. The default value is 16.

  Increasing the number of windows results in accurate averaging and smoother PSD waveforms. However, increasing the number of windows should usually be accompanied by increasing the Transient Stop Time for better accuracy.
- *PSD Summary*—This section provides a brief summary for the values that are used in generating the PSD waveforms. These values are optimized based on the Fourier

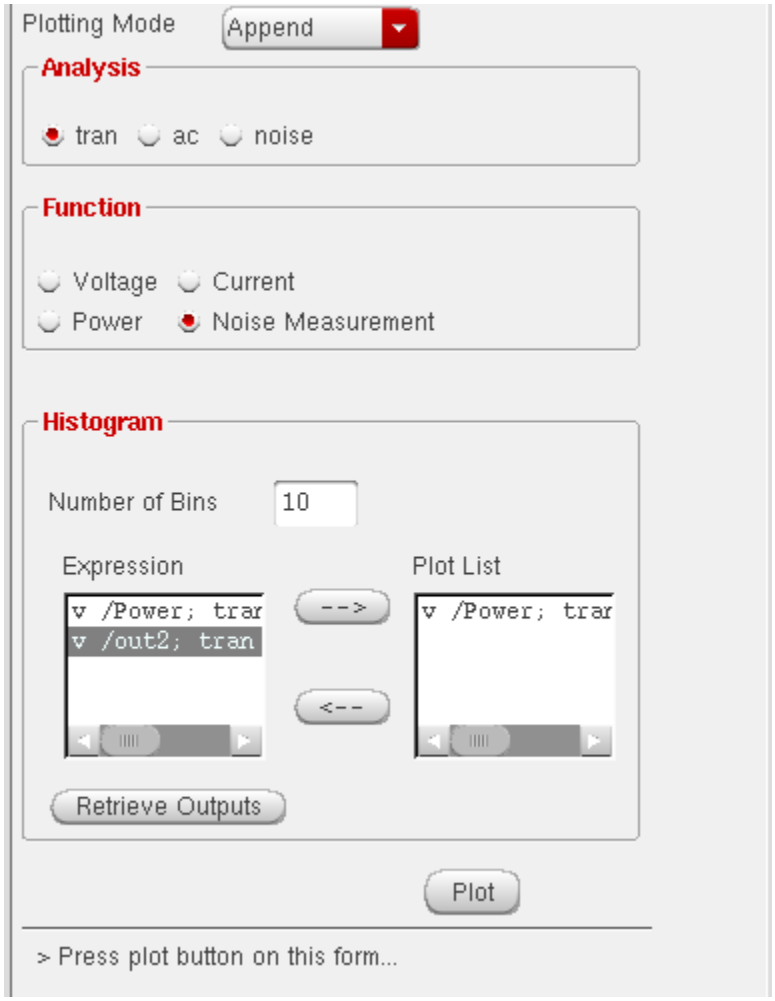

© 2023 Cadence Design Systems, Inc.

analysis inputs to minimize numerical errors in Fourier analysis, and thus, produce accurate PSD waveforms.

For more information, see the *Fourier Analysis* section in the *Spectre Classic Simulator, Spectre APS, Spectre X, and Spectre XPS User Guide*.

### Histogram Plots for Transient Noise Analysis

Transient noise analysis is displayed via histogram plots. The *Direct Plot* form corresponding to the transient noise analysis displays a *Histogram* option.




**Note:** Gaussian distribution of random noise sources in transient noise analysis is truncated at 3 sigma value, because infinite noise values would cause numerical problems. It can be observed in noise histogram of an ideal single resistor circuit. It has no effect on average


© 2023 Cadence Design Systems, Inc.

noise measurements and noise distribution for any physical circuit with band limited transfer function and multiple noise sources.

- To plot histograms for the selected waveform measurement(s), click *Plot*.

  **Note:** You must create the waveform expressions in the ADE window before plotting a histogram for any waveform measurement.

- To add specific waveform measurements to the *Plot List*, select the required waveform measurement in the *Expression* list and use the right-arrow button to add it to the *Plot List.*

- To remove specific waveform measurements from the *Plot List*, select the required waveform measurement in the *Plot List* and use the left-arrow button to put it back in the *Expression* list.

- If you create additional waveform expressions (through Calculator) after the *Direct Plot* form has been launched, you can click *Retrieve Outputs* to import all the existing waveform expressions into the ADE window.

  **Note:** There is no change in the working of all normal plots and measurements.


© 2023 Cadence Design Systems, Inc.

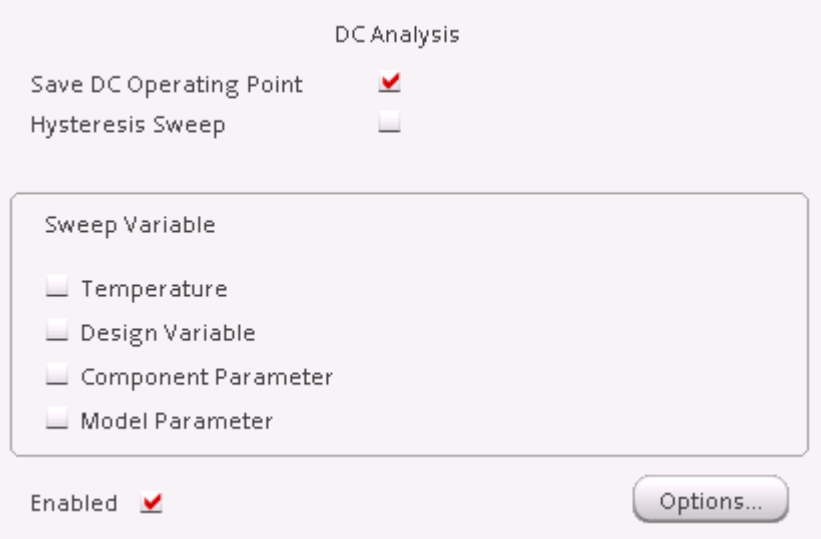
## DC Analysis

DC analysis finds the DC operating point or DC transfer curves of the circuit. To generate transfer curves, specify a parameter and a sweep range. The parameter can be a temperature, a design variable, a device instance parameter, or a device model parameter.

Perform the following steps to set up a DC analysis:

1. In the *Choosing Analyses* form, choose *dc*.

   The *Choosing Analyses* form is updated to show the additional fields for DC analysis.



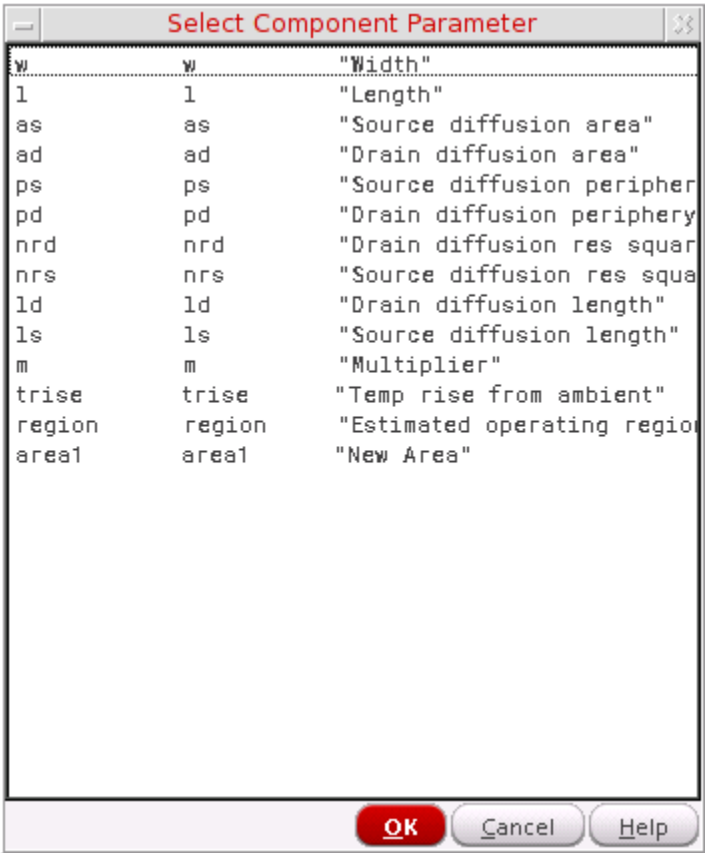

2. Select the *Save DC Operating Point* check box if you want to save the DC operating point information.
3. Select the *Hysteresis Sweep* check box if you want to enable DC hysteresis sweep.
4. Choose a variable to sweep during DC analysis, in the *Sweep Variable* section.
   - To sweep circuit temperature, select the *Temperature* check box.
   - To sweep a design variable, select the *Design Variable* check box, then specify the name of the design variable in the *Variable Name* field, or click the *Select Design Variable* button to select the design variable.
   - To sweep a device instance parameter, select the *Component Parameter* check box, then click *Select Component*.

     The schematic for the design is displayed, perform the following steps to select the parameter:


© 2023 Cadence Design Systems, Inc.

- Click the instance whose parameter you want to sweep.

  The *Select Component Parameter* form appears.



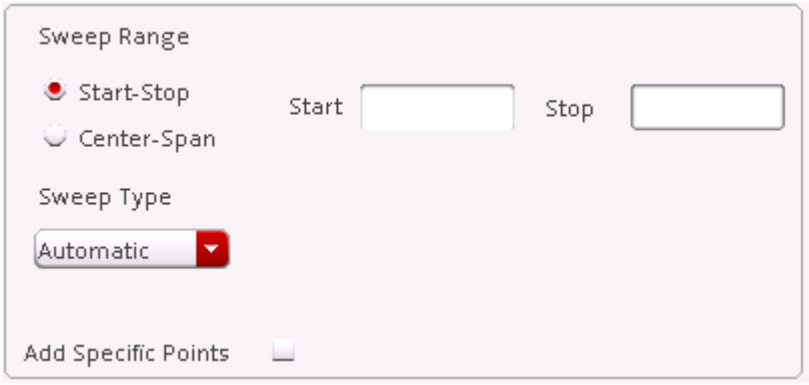

- Select the parameter you want to sweep and click *OK*.

The instance name of the component is displayed in the *Component Name* field and the parameter name is displayed in the *Parameter Name* field.

- To sweep a model parameter, select the *Model Parameter* check box, then specify the model name in the *Model Name* field and the parameter name in the *Parameter Name* field.


© 2023 Cadence Design Systems, Inc.

**5.** Perform the following steps to specify the sweep range for the sweep variables:




**a.** Perform one of the following steps to specify the *Sweep Range*:

- Select the *Start-Stop* option and specify the start sweep limit in the *Start* field and the stop sweep limit in the *Stop* field.
- Select the *Center-Span* option and specify the center of the sweep in the *Center* field and the sweep limit span limit in the *Span* field.

**b.** Select the sweep type from the *Sweep Type* drop-down list.

If the sweep type is *Linear*, perform one of the following steps:

- Select the *Step Size* option and specify the step size for the linear sweep.
- Select the *Number of Steps* option and specify the number of steps for the linear sweep.

If the sweep type is *Logarithmic*, perform one of the following steps:

- Select the *Points Per Decade* option and specify the points per decade for the logarithmic sweep.
- Select the *Number of Steps* option and specify the number of steps for the logarithmic sweep.

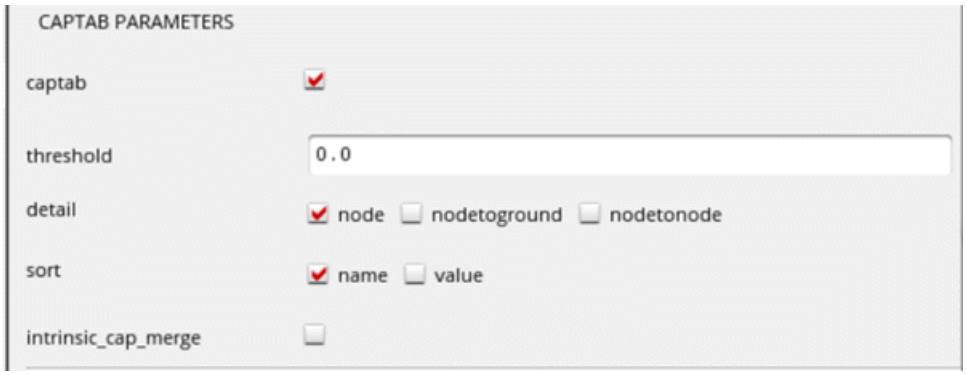
**c.** Optionally, select the *Add Specific Points* check box to specify a list of values to sweep. Use spaces to separate each value in the field.

**d.** Optionally, select the *Add Points By File* check box to choose the file containing the values to sweep. You can either enter the path to the file in the text box manually, or click the *Browse* button next to the text box and choose the file.


© 2023 Cadence Design Systems, Inc.

### CAPTAB Parameters

You can generate capacitive loading information about a circuit after a Spectre simulation. The following additional components are available on the *DC Options* form:




- *captab*—Indicates you have enabled or disabled captab parameters. Select the *captab* check box to enable captab parameters.
- *threshold*—Indicates the threshold value in real numbers. Results below this value are omitted from the output. The default value is `0.0`.
- *detail*—Can be set to `node`, `nodetoground`, and `nodetonode`. The default value is `node`.
- *sort*—Can be set to `name` and `value`. Indicates how to sort the entries in the table; according to value or alphabetically, by name. The default value is `name`.
- *intrinsic_cap_merge*—Merges the internal captab node with the external node. Possible values are `yes` and `no`. The default value is `no`.

  You can also specify this parameter in the *Name* column of the *Save circuit information analysis* group box in the Save Options form. This is possible only when the *What* column is set to *captab*.

For details on DC Analysis refer to the *Analysis Statements* chapter of the *Spectre Circuit Simulator Reference*.

### Sweeping a Variable

Perform the following steps to run a DC Transfer Curve analysis and sweep a variable:

1. Choose a sweep variable.


© 2023 Cadence Design Systems, Inc.

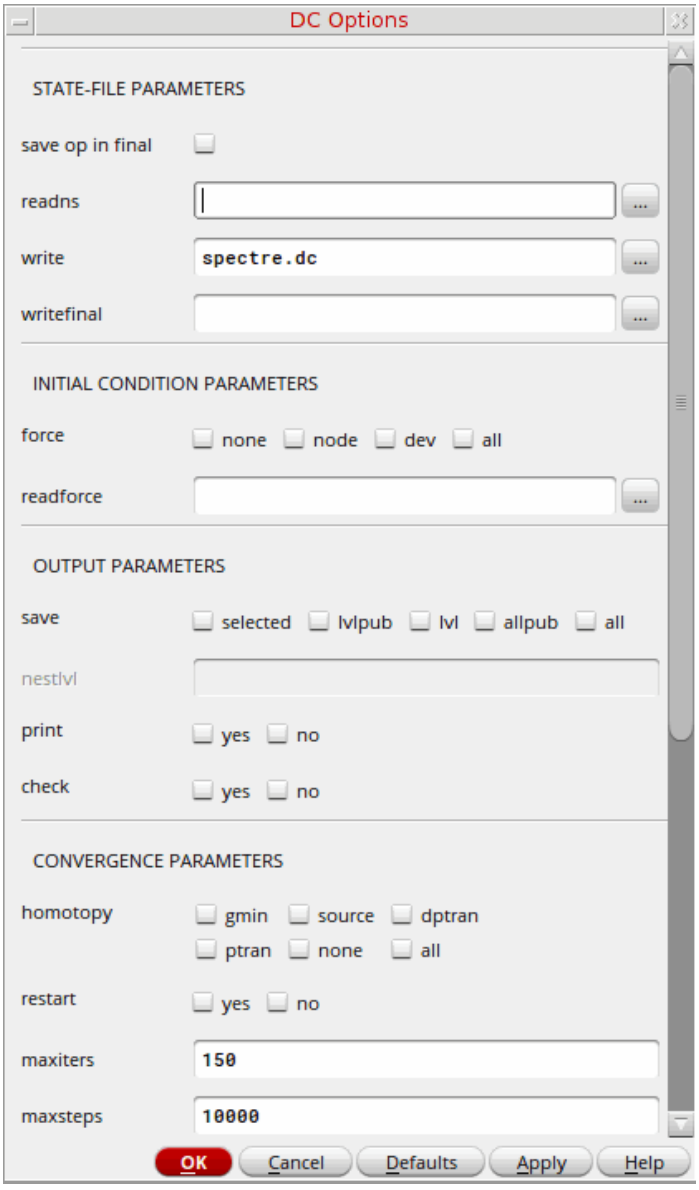
The *Choosing Analyses* form is updated to show the additional fields.

2. Specify the necessary parameters.
   - To sweep a design variable, fill out the name of the design variable, or click *Select Design Variable* to choose from the list box.
   - To sweep a component, specify the component name and the parameter to sweep. Click *Select Component* to select the component in the schematic.
   - To sweep a model parameter, enter the model and parameter names.
3. Specify the sweep range and sweep type.

   The sweep type options are mapped to Spectre statements:
   - Linear + Step Size = step
   - Linear + Number of Steps = lin
   - Logarithmic + Points Per Decade = dec
   - Logarithmic + Number of Steps = log
   - Add Specific Points = values=[…]
4. Click *Options* to set the options controlling DC simulation.


© 2023 Cadence Design Systems, Inc.

The *DC Options* form appears.




For more information about the options in the form, see the *DC Analysis* section in the *Analysis Statements* chapter of the *Spectre Circuit Simulator Reference*.

**Note:** If a Spectre option does not have a GUI equivalent in the Virtuoso ADE forms, you can specify these options in the *additionalParams* field on the options form for


© 2023 Cadence Design Systems, Inc.

analyses.

5. Click *Apply*.


© 2023 Cadence Design Systems, Inc.

## AC Analysis

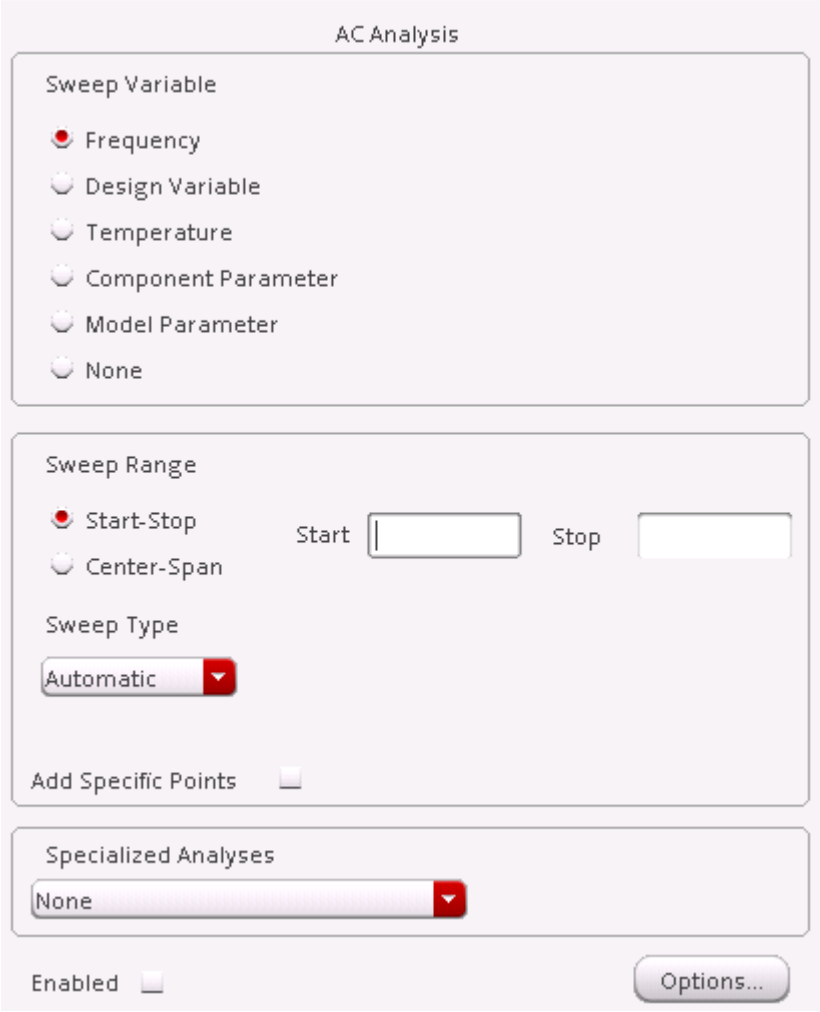
AC analysis linearizes the circuit about the DC operating point and computes the response to a given small sinusoidal stimulus. Spectre can perform the analysis while sweeping a parameter.

The parameter can be a frequency, a design variable, temperature, a component instance parameter, or a component model parameter. If changing a parameter affects the DC operating point, the operating point is recomputed on each step.

Perform the following steps to set up the an AC analysis:

1. In the *Choosing Analyses* form, choose *ac*.

   The *Choosing Analyses* form is updated to show the additional fields for AC analysis.





© 2023 Cadence Design Systems, Inc.

2. Choose a sweep variable and specify any necessary parameters.

   - If you do not sweep the frequency, specify the frequency at which to sweep the variable.
   - If you sweep a design variable, fill out the name of the design variable, or select from the list box after clicking the select button.
   - If you sweep a component, specify the parameter to sweep. Click *Select Component* to specify the component using the schematic.
   - If you sweep a model parameter, enter the model and parameter names.

3. Specify the sweep range and sweep type.

   Enter the start and stop points of the range or the center and span of the range.

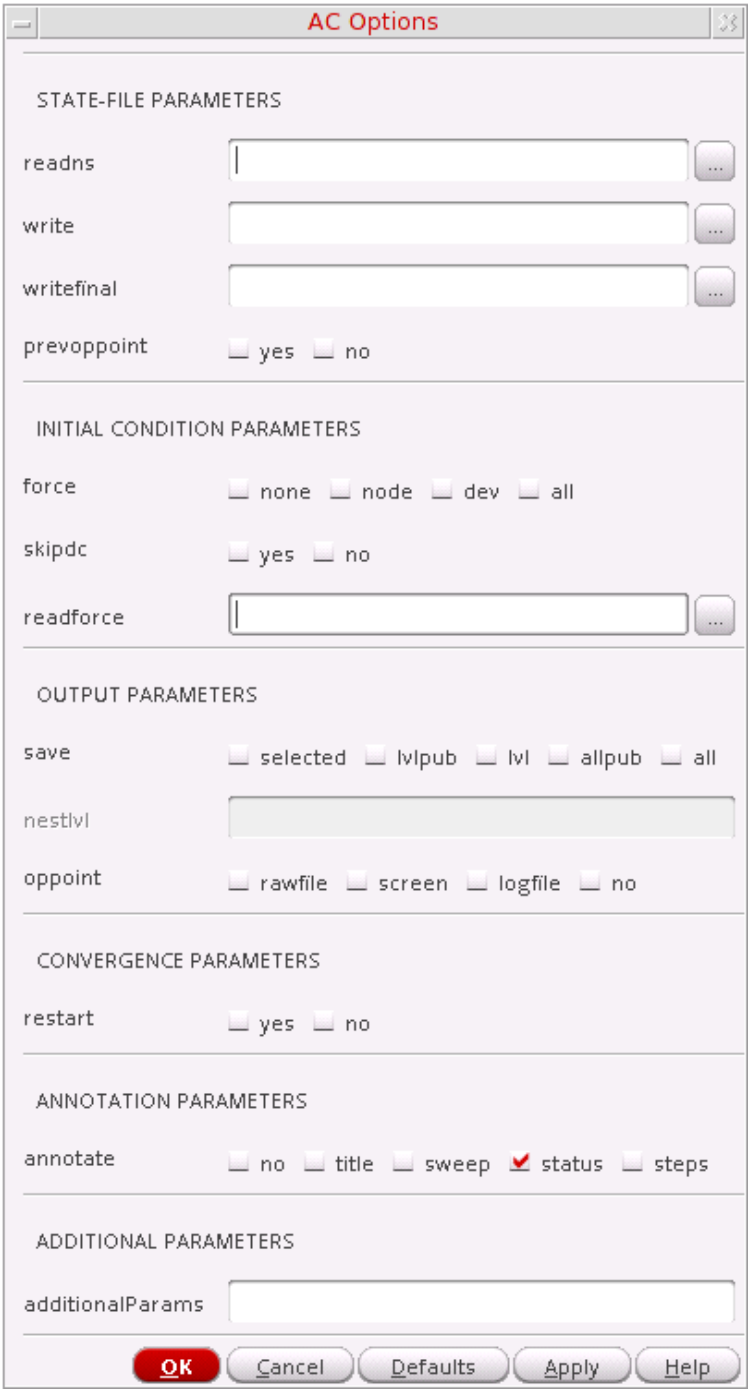
   The sweep type options are mapped to Spectre statements:

   - Linear + Step Size = step
   - Linear + Number of Steps = lin
   - Logarithmic + Points Per Decade = dec
   - Logarithmic + Number of Steps = log
   - Add Specific Points = values=[…]
   - Add Points By File = valuesfile

4. Click *Options* to select the Spectre options controlling the simulation.


© 2023 Cadence Design Systems, Inc.

The *AC Options* form appears.




For more information about the options in the form, see the *AC Analysis* section in the *Analysis Statements* chapter of the *Spectre Circuit Simulator Reference*.

5. Select the *Enabled* check box and click *Apply*.


© 2023 Cadence Design Systems, Inc.

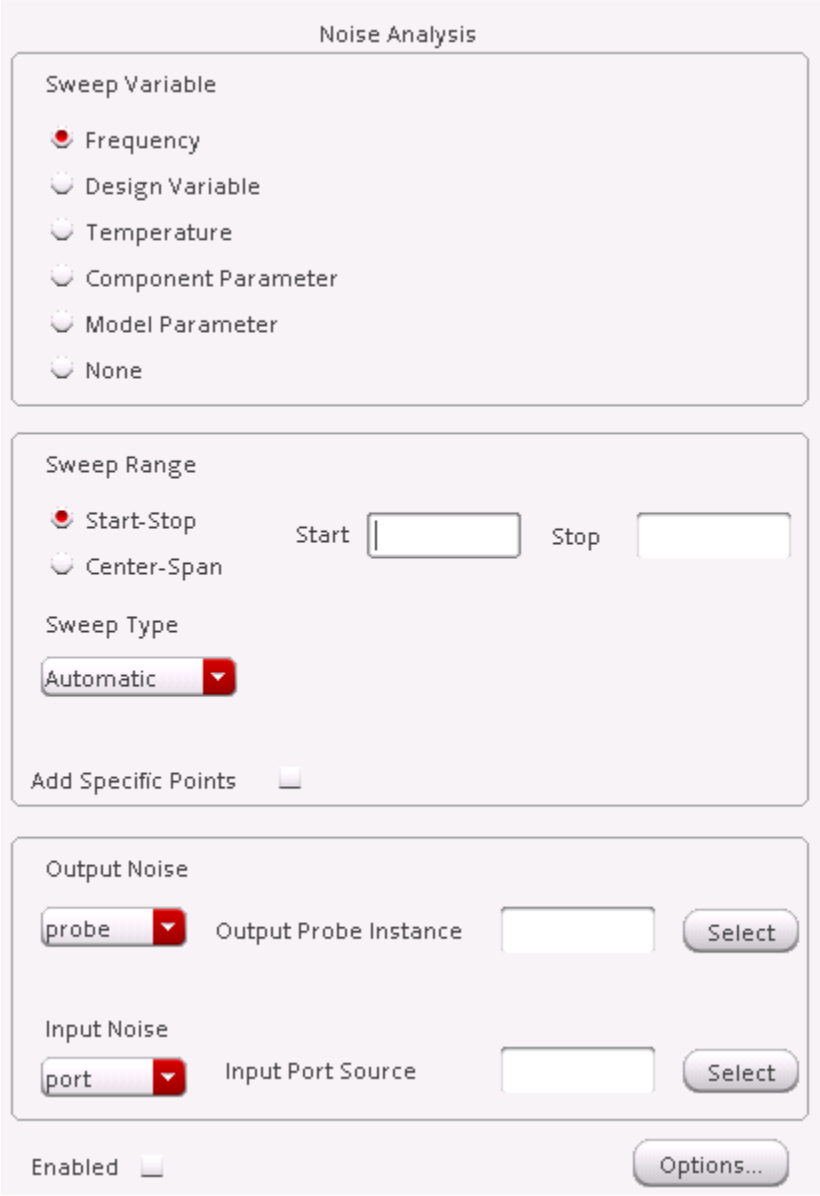
## Noise Analysis

Noise analysis linearizes the circuit about the DC operating point and computes the total-noise spectral density at the output. If you specify an input probe, the transfer function and the input-referred noise for an equivalent noise-free network is computed.

Perform the following steps to set up a noise analysis:

1. In the *Choosing Analyses* form, choose *noise*.

   The *Choosing Analyses* form is updated to show the additional fields for noise analysis.

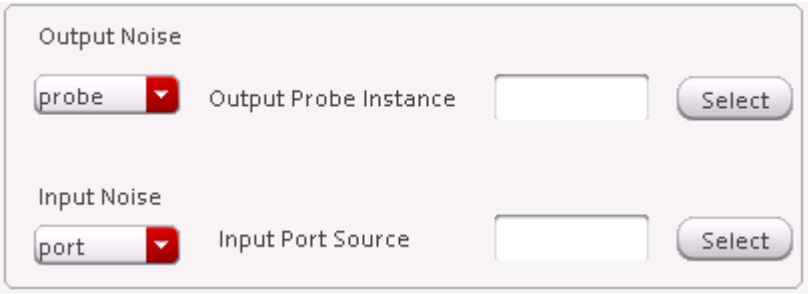




© 2023 Cadence Design Systems, Inc.

2. Choose a sweep variable and specify any necessary parameters.
   - If you do not sweep the frequency, specify the frequency at which to sweep the variable.
   - If you sweep a design variable, fill out the name of the design variable, or choose from the list box after pressing the select button.
   - If you sweep a component, specify the analysis frequency, component name, and the parameter to sweep. Click *Select Component* to specify the components using the schematic.
   - If you sweep a model parameter, enter the model and parameter names.
3. Specify the sweep range and sweep type.

   The sweep type options are mapped to Spectre statements:
   - Linear + Step Size = step
   - Linear + Number of Steps = lin
   - Logarithmic + Points Per Decade = dec
   - Logarithmic + Number of Steps = log
   - Add Specific Points = values=[…]
   - Add Points By File = valuesfile
4. Choose an *Output Noise* option.




   - To measure the output noise voltage, choose *voltage* in the *Output Noise* drop-down list, and specify values for *Positive Output Node* and *Negative Output Node*, and click a net in the schematic.
   - To measure the output noise probe, choose *probe* in the *Output Noise* drop-down list and click *Select* opposite *Output Probe Instance*, and click a voltage source in the schematic.


© 2023 Cadence Design Systems, Inc.

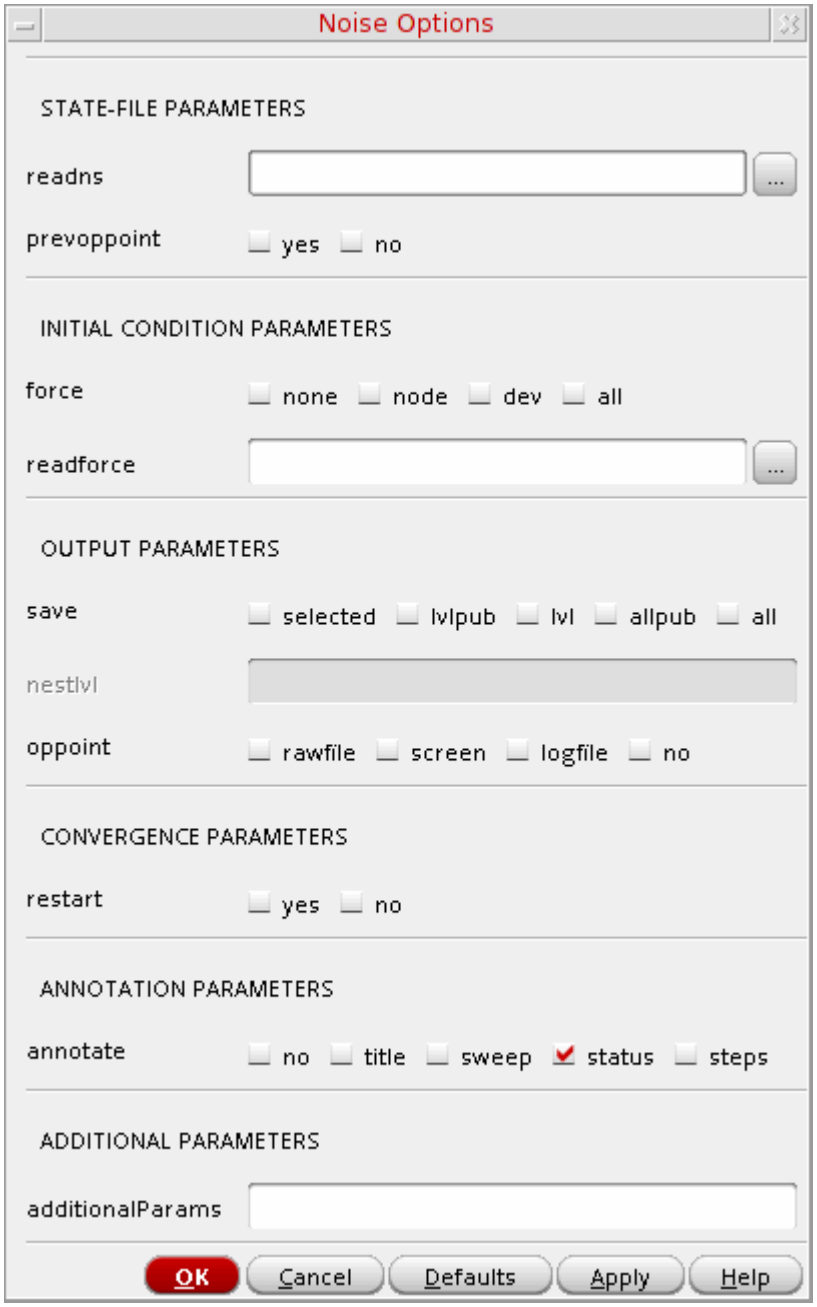
**Note:** While selecting nodes, select the nodes/nets around the desired instance.

5. Optionally, choose an *Input Noise* option.
   - Choose *voltage*, *current*, or *port*.
   - Click *Select* to choose the *Input Voltage Source* or *Input Current Source* or *Input Port Source.*
   - Click a source or port in the schematic.
   - Click *Apply*.
6. If you want to separate the noise into noise sources and transfer functions, select the Noise Separation check box.

7. Click *Options* to set the Spectre options controlling noise simulation.


© 2023 Cadence Design Systems, Inc.

The *Noise Options* form appears.




For more information about the options in the form, see the *Noise Analysis* section in the *Analysis Statements* chapter of the *Spectre Circuit Simulator Reference*.

**8.** Select the *Enabled* check box and click *Apply*.


© 2023 Cadence Design Systems, Inc.

## S-parameter Analysis

S-parameter analysis linearizes the circuit about the DC operating point and computes S-parameters of the circuit taken as an N-port. The `psin` instances (netlist-to-Spectre port statements) define the ports of the circuit. Each active port is turned on sequentially, and a linear small-signal analysis is performed. The Spectre simulator converts the response of the circuit at each active port into S-parameters and prints these parameters. There must be at least one active port (analogLib `psin` instance) in the circuit.

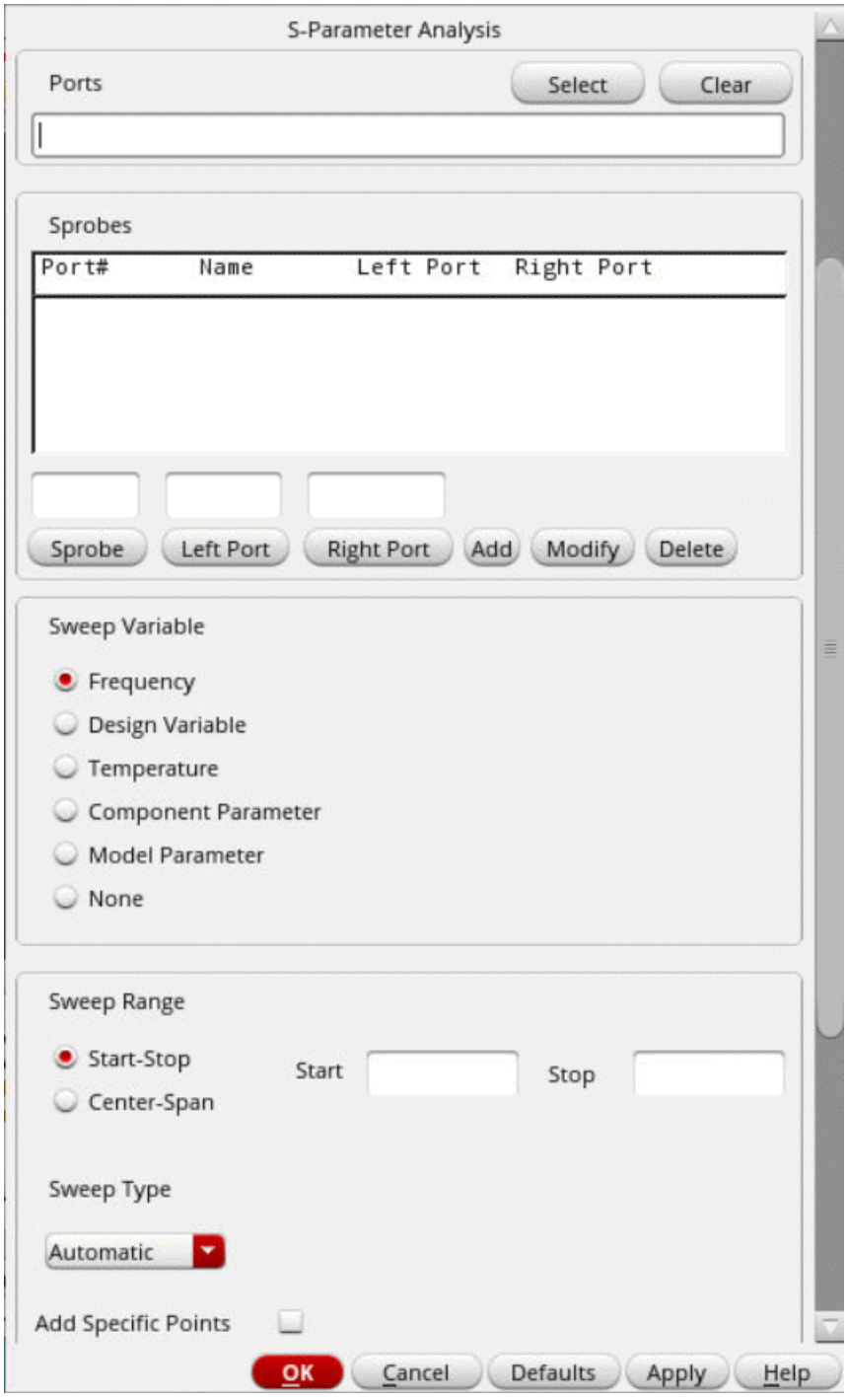
The parameter can be a frequency, a design variable, temperature, a component instance parameter, or a component model parameter. If changing a parameter affects the DC operating point, the operating point is recomputed on each step.

Perform the following steps to set up an S-parameter analysis:

1. In the *Choosing Analyses* form, choose *sp*.


© 2023 Cadence Design Systems, Inc.

The *Choosing Analyses* form is updated to show the additional fields for S-parameter analysis.




2. Specify the list of active ports in the *Ports* field.


© 2023 Cadence Design Systems, Inc.

In this field, the ports are numbered sequentially, beginning with one, in the order given. Otherwise, all ports present in the circuit are active and the port numbers used are those that were assigned on the port statements.

3. In the *Sprobes* group box,

   a. Click *Sprobe* to select an sprobe from the schematic or manually add the sprobe name in the field.

   b. Click *Left Probe* to specify the left probe from the schematic or manually add the probe name in the field.

   c. Click *Right Probe* to specify the right probe from the schematic or manually add the probe name in the field.

   d. Click *Add* to add the specified sprobe to the *Sprobes* table.

   e. Click *Modify* to modify the selected sprobe entry in the table.

   f. Click *Delete* to delete the selected sprobe entry from the table.

4. Choose a sweep variable and specify any necessary parameters.

   - If you do not sweep the frequency, specify the frequency at which to sweep the variable.
   - If you sweep a design variable, fill out the name of the design variable, or select from the list box after clicking the select button.
   - If you sweep a component, specify the parameter to sweep. Click *Select Component* to select the component in the schematic.
   - If you sweep a model parameter, enter the model and parameter names.

5. Specify the sweep range and sweep type.

   Enter the start and stop points of the range or the center and span of the range.

   The sweep type options are mapped to Spectre statements:

   - Linear + Step Size = step
   - Linear + Number of Steps = lin
   - Logarithmic + Points Per Decade = dec
   - Logarithmic + Number of Steps = log
   - Add Specific Points = values=[…]

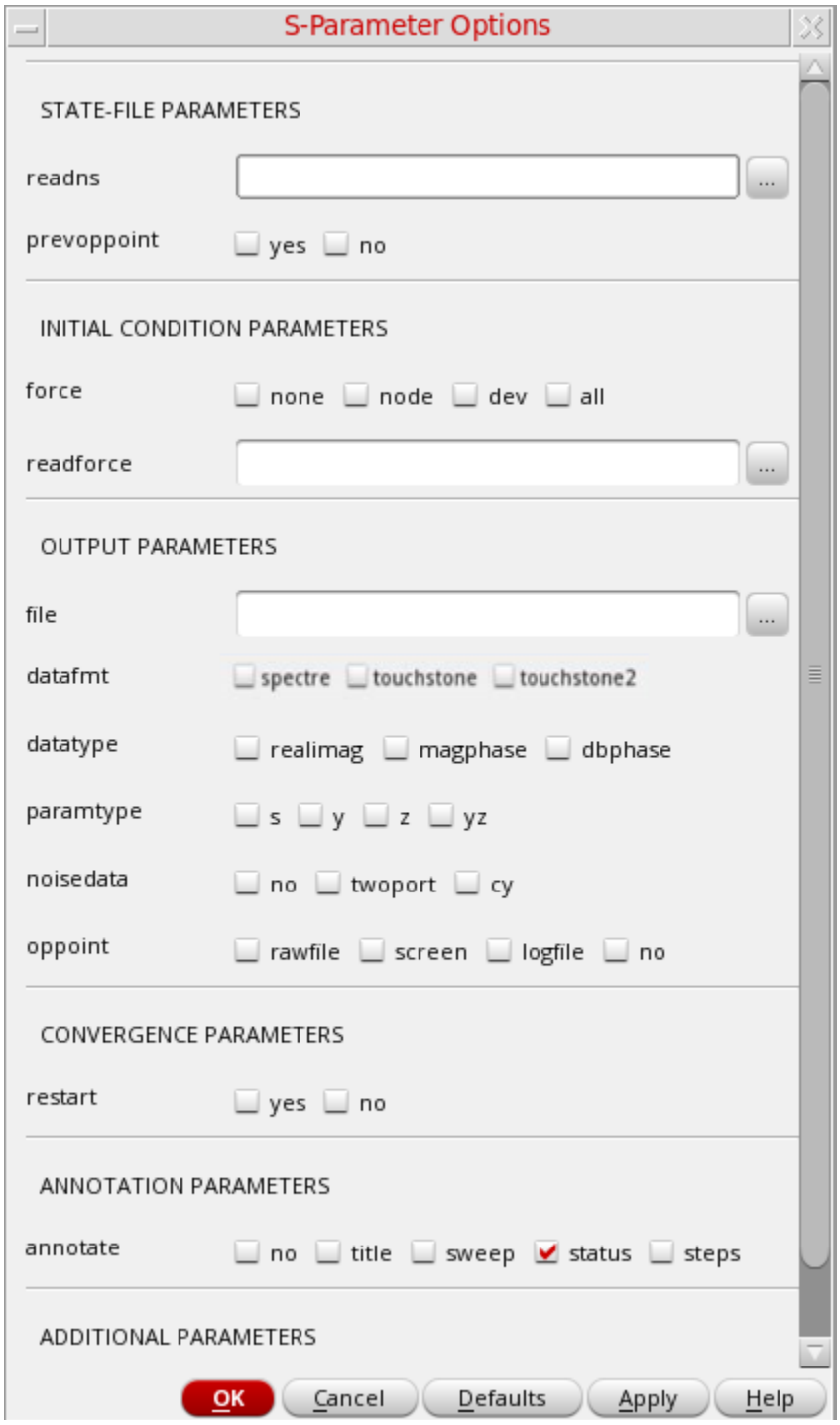

© 2023 Cadence Design Systems, Inc.

- Add Points By File = valuesfile

6. From the Mode section, select one of the following s-parameter analysis mode:
   - *Single-Ended*: allows the plotting or simulation of single-ended s-parameters.
   - *Mixed In/Out*: allows the plotting or simulation of mixed-mode s-parameters.
   - *Other*: allows the plotting or simulation of a combination of single-ended and mixed-mode s-parameters.
7. Click *Options* to select the Spectre options controlling the simulation.


© 2023 Cadence Design Systems, Inc.

The *S-parameter Options* form appears.




In the OUTPUT PARAMETERS section,


© 2023 Cadence Design Systems, Inc.

- If you select *touchstone* in the datafmt field, the *cy* check box in the noisedata field is disabled.
- If you select *spectre* in the datafmt field, the *twoport* check box in the noisedata field is disabled.
- If you select *touchshone* in the *datafmt* field and you have specified more than two ports in the *Ports* field in the S-Parameter Analysis section of the Choosing Analyses form, all three check boxes in the *noisedata* field, *no*, *twoport*, and *cy* are disabled.

**Note:** You can see the option *touchstone2* in the *datafmt* field from SPECTRE17.1 release onwards. If you select *touchstone2*, it is netlisted and the output file is created in version 2 format.

For more information about the options in the form, see the *S-parameter Analysis* section in the *Analysis Statements* chapter of the *Spectre Circuit Simulator Reference*.

8. In the *Do Noise* section, select the *yes* check box to perform Noise analysis.
9. Select the *Enabled* check box and click *Apply*.

### Mixed-Mode S-parameter Analysis

Mixed-mode S-parameters are used for the analysis of differential circuits and provide the ability to analyze the signal flow through differential or balanced lines.

There are four types of mixed-mode S-parameters.

- Sdd: differential-mode S-parameters.
- Scd: common-to-differential mode S-parameters.
- Sdc: differential-to-common mode S-parameters.
- Scc: common-mode S-parameters.

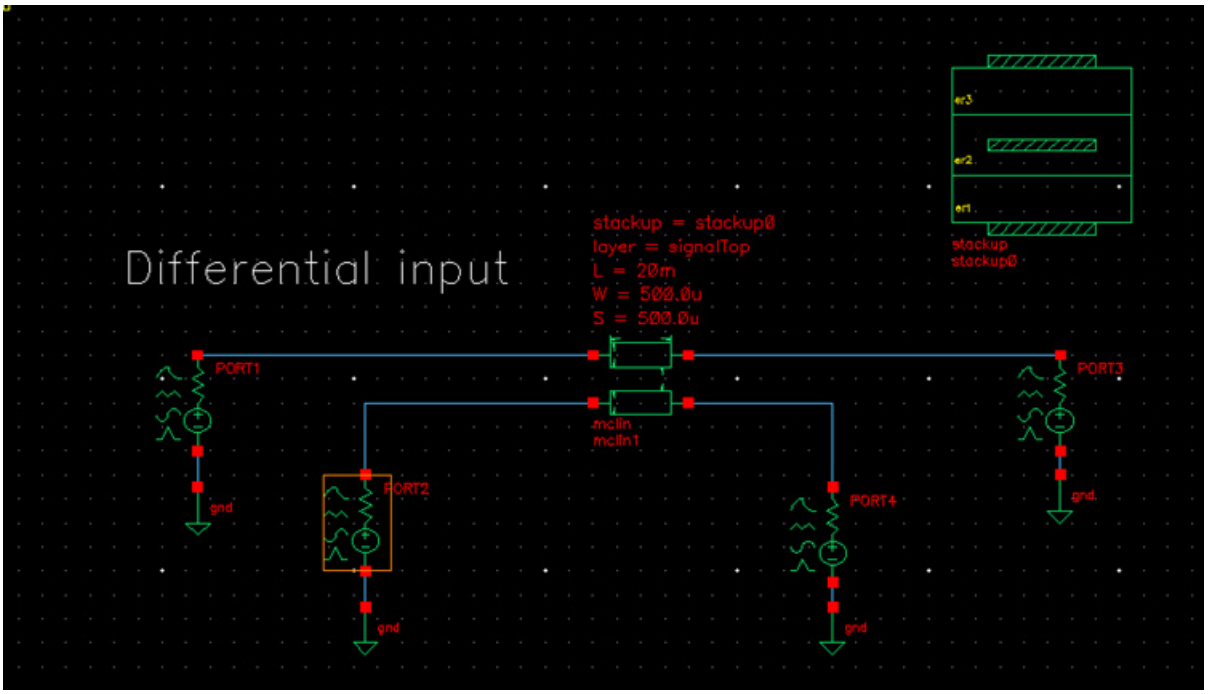

© 2023 Cadence Design Systems, Inc.

They can be defined as:

$$\begin{bmatrix} b_{d1} \\ b_{d2} \\ b_{c1} \\ b_{c2} \end{bmatrix} = \begin{bmatrix} s_{dd11} & s_{dd12} & s_{dc11} & s_{dc12} \\ s_{dd21} & s_{dd22} & s_{dc21} & s_{dc22} \\ s_{cd11} & s_{cd12} & s_{cc11} & s_{cc12} \\ s_{cd21} & s_{cd22} & s_{cc21} & s_{cc22} \end{bmatrix} \begin{bmatrix} a_{d1} \\ a_{d2} \\ a_{c1} \\ a_{c2} \end{bmatrix} = S_{mm} \begin{bmatrix} a_{d1} \\ a_{d2} \\ a_{c1} \\ a_{c2} \end{bmatrix} = \begin{bmatrix} S_{dd} & S_{dc} \\ S_{cd} & S_{cc} \end{bmatrix} \begin{bmatrix} a_{d1} \\ a_{d2} \\ a_{c1} \\ a_{c2} \end{bmatrix}$$

Let us consider an example to understand how to set up a mixed-mode S-parameter analysis.

The following schematic represents a simple testbench used to measure the characteristics of a section of a coupled transmission line. It consists of input ports, output ports, an mclin component from rfTLib, and a stackup.




This is a simple testbench used to measure the characteristics of a section of a coupled transmission line.

Set up two input ports, PORT1 and PORT2. Their port numbers are 1 and 2 respectively.

**Note:** For transient analysis, the source type would be set to sine, pulse, pwl, etc but since this is a small signal sp analysis, you need to select the *Display small signal params* check box and specify the AC Magnitude (Vpk) and AC Phase.

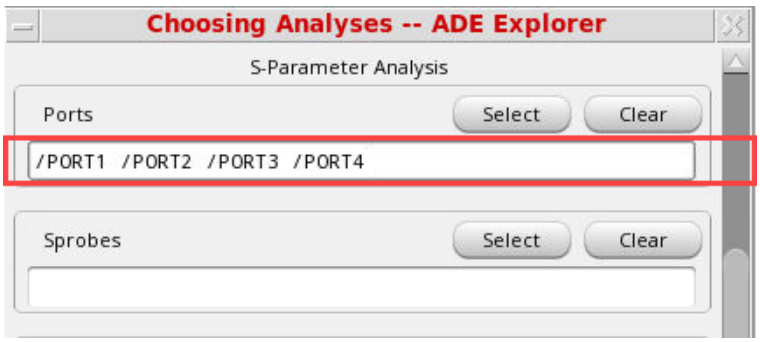

© 2023 Cadence Design Systems, Inc.

Set up the input to be differential and the reference resistance for both ports as `50 ohms`.

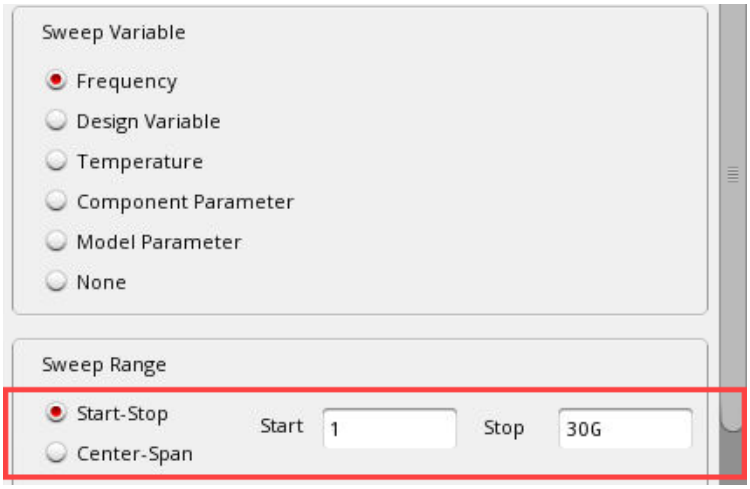
Now, set up two output loads, PORT3 and PORT4. Their port numbers are 3 and 4 respectively. Set the source type to dc and the output reference resistance as `50 ohms`.

Let us now look at the mixed-mode measurement setup.

1. In the *Choosing Analysis* form, select *sp*.
2. In the schematic, select two input ports and two output ports.



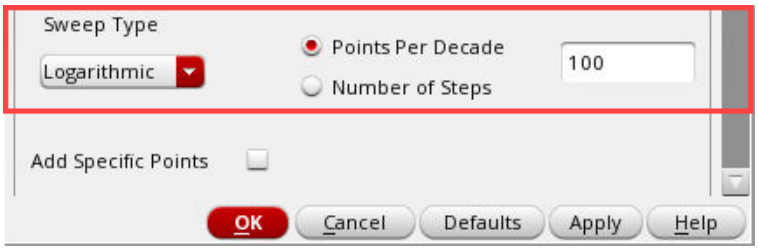

3. Set the frequency sweep range as `1` to `30G`.





© 2023 Cadence Design Systems, Inc.

4. From the *Sweep Type* drop-down menu, select *Logarithmic* and set it to `100` points per decade.




5. Select the *Mixed In/Out* option.

6. Click *OK* and run a simulation.

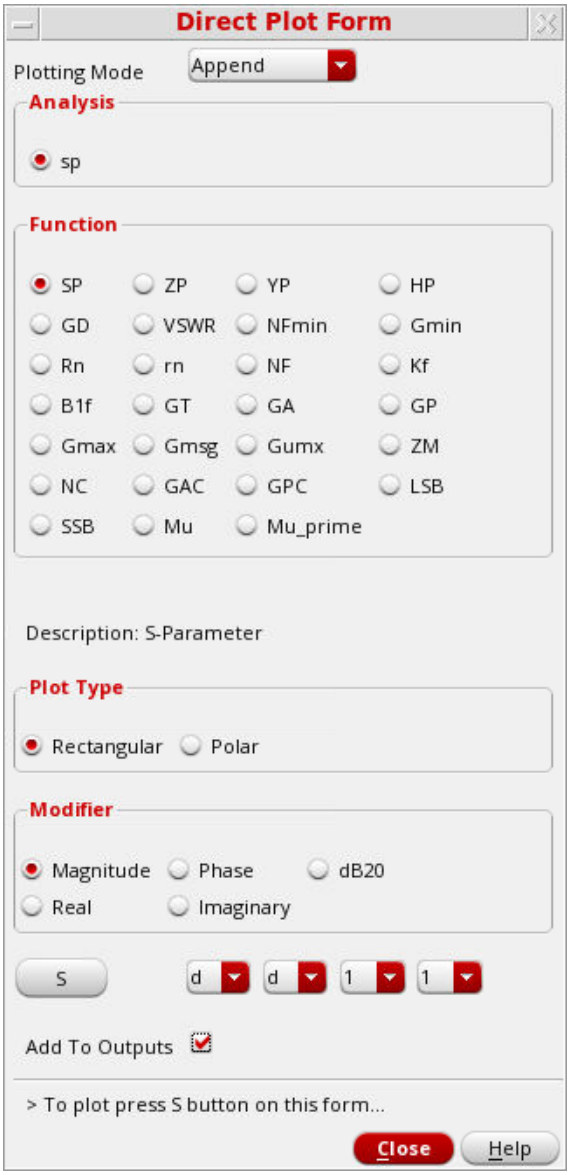
   Note the following

   - When the mode is set to *Mixed In/Out*, differential and common-mode S-parameters, denoted as mixed mode S-parameters are calculated. When simulating these mixed mode S-parameters, there must be 2N active port statements in the circuit, where N must be greater than one.
   - The mixed-mode S-parameters are calculated referring to the pairing of the ports, with the port numbers ordered in pairs as `(1,2) (3,4)`, and so on in the ports list. When the *Mixed IN/Out* mode is selected, spectre calculates the differential-to-differential, differential-to-common, common-to-differential, and common-to-common S-parameters.

7. Select *Results—Direct Plot—Main Form*.


© 2023 Cadence Design Systems, Inc.

The *Direct Plot* form is displayed.

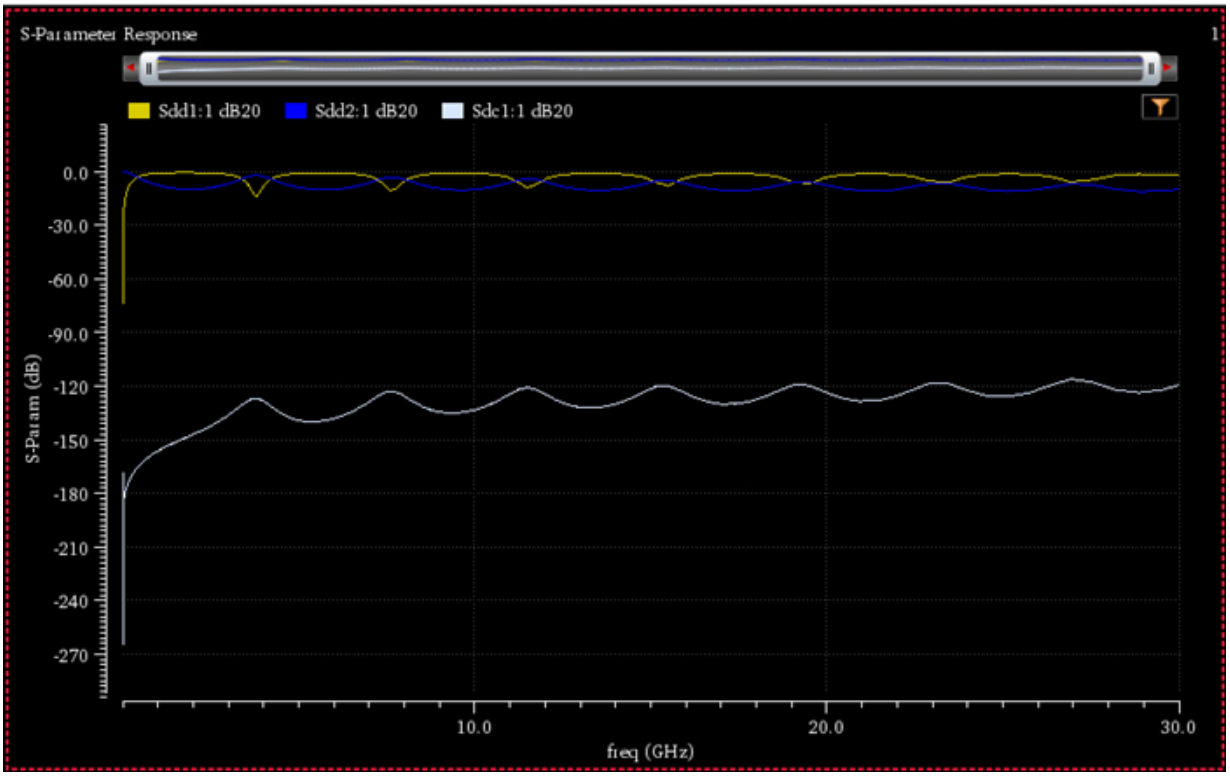



You can plot the mixed-mode S-parameters on a Rectangular or Polar plot.

8. In the Plot Type section, select *Rectangular*.
9. Select *dB20* as the modifier.
10. Choose the desired mixed-mode S-parameter and click *S*.


© 2023 Cadence Design Systems, Inc.

The following figure shows the plot in dB for Sdd11, Sdd21 and Sdc11.




**Note:** To view these results in the outputs setup, select the *Add to Outputs* check box in the *Direct Plot* Form.

| Name | Type | Details | Value | Plot | Save | Spec |
|---|---|---|---|---|---|---|
| Sdd1:1 dB20 | expr | db(spm('sp 1 1 ?port1 "d" ?port... | | ☑ | ☐ | |
| Sdd2:1 dB20 | expr | db(spm('sp 2 1 ?port1 "d" ?port... | | ☑ | ☐ | |
| Sdc1:1 dB20 | expr | db(spm('sp 1 1 ?port1 "d" ?port... | | ☑ | ☐ | |

The expressions are then saved in the maestro outputs setup and can be automatically plotted the next time you run a simulation.


© 2023 Cadence Design Systems, Inc.

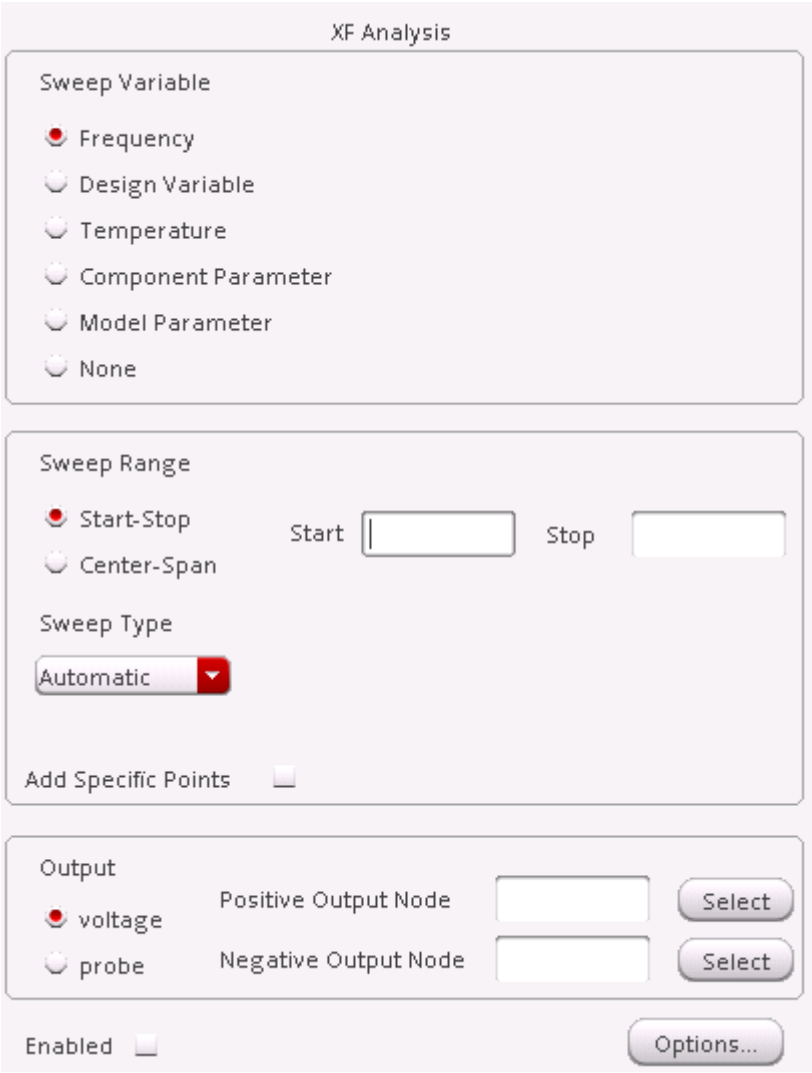
## Transfer Function Analysis

Transfer function, or *xf*, analysis linearizes the circuit about the DC operating point and performs a small-signal analysis that calculates the transfer function from every independent source or instance terminal in the circuit to a designated output. The variable of interest at the output can be voltage or current.

Perform the following steps to set up a transfer function analysis:

1. In the *Choosing Analyses* form, choose *xf*.

   The *Choosing Analyses* form is updated to show the additional fields for transfer function analysis.





© 2023 Cadence Design Systems, Inc.

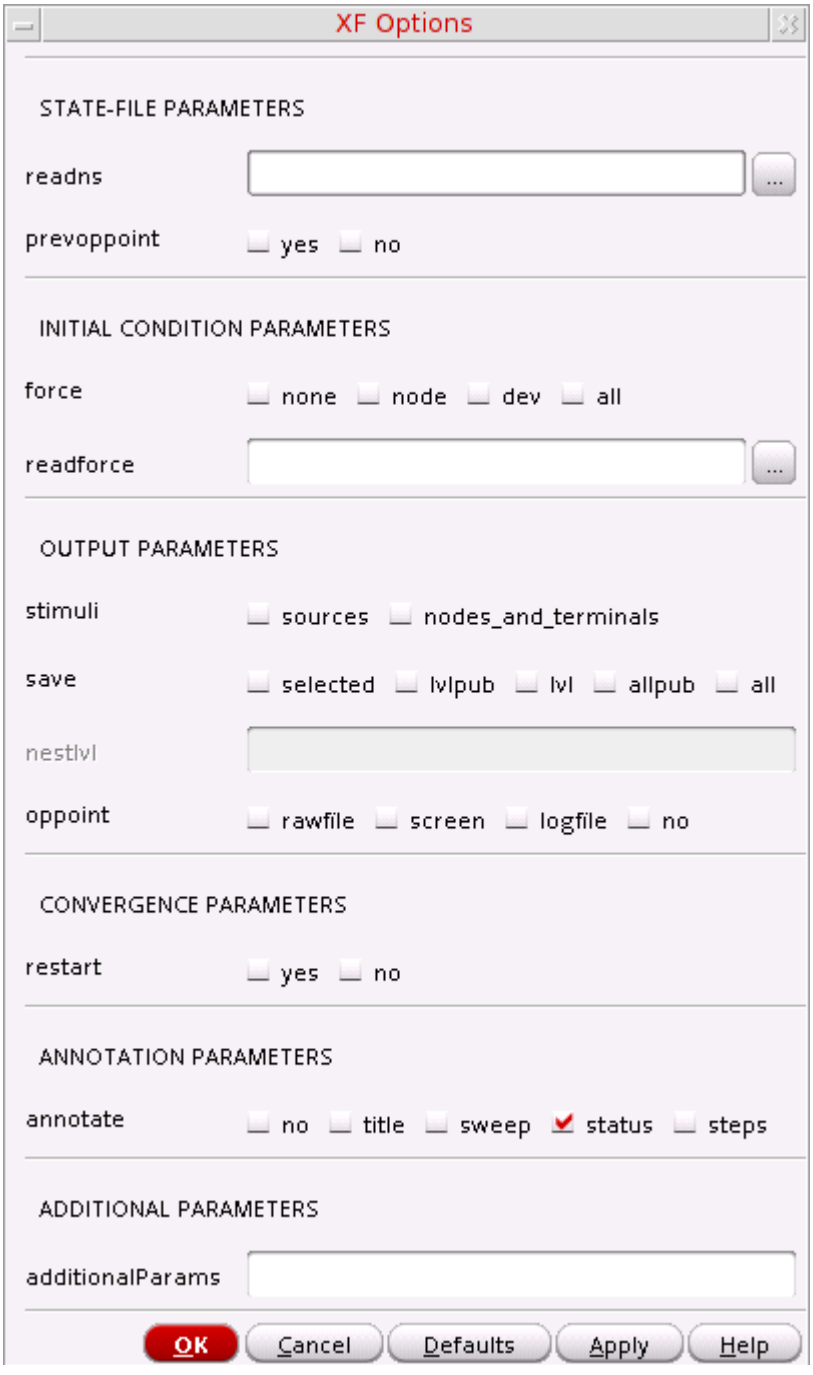
2. Select a sweep variable option and specify any necessary parameters.
   - If you do not sweep the frequency, specify the frequency at which to sweep the variable.
   - If you sweep a design variable, fill out the name of the design variable, or select from the list box after clicking the select button.
   - If you sweep a component, specify the analysis frequency, component name, and the parameter to sweep. Click *Select Component* to specify the components using the schematic.
   - If you sweep a model parameter, enter the model and parameter names.
3. Specify the sweep range and sweep type.

   The sweep type options are mapped to Spectre statements:
   - Linear + Step Size = step
   - Linear + Number of Steps = lin
   - Logarithmic + Points Per Decade = dec
   - Logarithmic + Number of Steps = log
   - Add Specific Points = values=[…]
   - Add Points By File = valuesfile
4. Choose *voltage* or *probe* for *Output*.
   - To measure the output voltage, click *Select* opposite *Positive Output Node* and click a net in the schematic.
   - To measure the output probe, click *probe*, click *Select* opposite *Output Probe Instance,* and click an instance in the schematic.

   **Note:** While selecting nodes, select the nodes/nets around the desired instance.
5. Click *Options* to set the Spectre options controlling transfer function simulation.


© 2023 Cadence Design Systems, Inc.

The *XF Option* form appears.




6. Set the options as needed and click *Apply*.


© 2023 Cadence Design Systems, Inc.

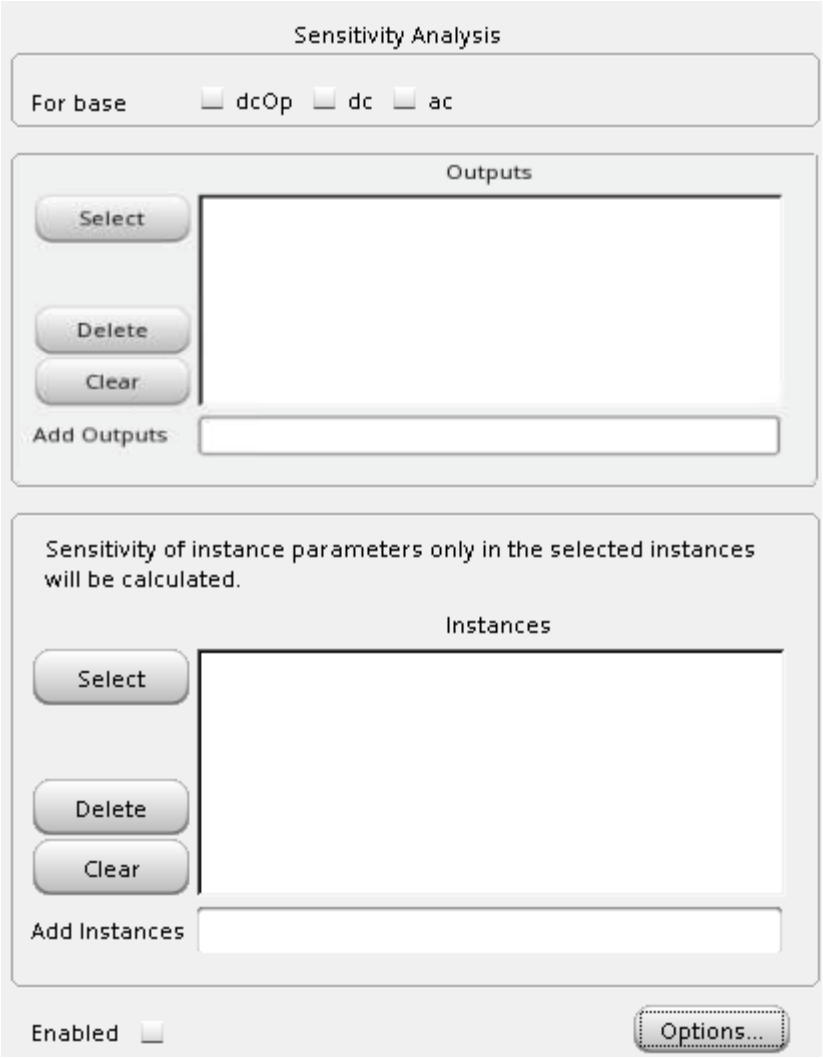
## Sensitivity Analysis

Sensitivity analysis helps a designer see which parameters in a circuit most affect the specified outputs. It is typically used to tune a design to increase or decrease certain design goals. You might run a sensitivity analysis to determine which parameters to optimize using the optimizer.

Perform the following steps to set up a sensitivity analysis:

1. In the *Choosing Analyses* form, choose *sens*.

   The *Choosing Analyses* form is updated to show the additional fields for sensitivity analysis.




2. Choose the types of sensitivities you want to calculate.


© 2023 Cadence Design Systems, Inc.

In the *For base* field, choose any of the analyses on which you want to perform a sensitivity analysis. The available analyses are *dcOp* (DC operating point), *dc*, and *ac*.

**Note:** Before you run a sensitivity analysis, you must run the corresponding base analysis.

3. In the *Outputs* section, click *Select* to choose the output objects you want to measure.

   The schematic opens in a new tab. Use the *Escape* key to end selection.

   **Note:** The schematic must be open before you can select any outputs. If you do not select a valid output object, a warning stating that the object selected is not a valid selection object is displayed in the CIW.

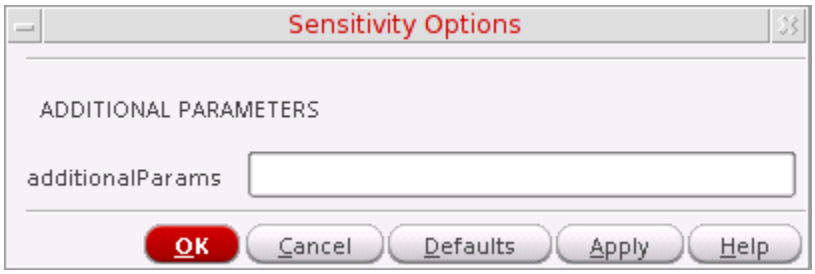
4. In the *Instances* section, click *Select* to choose the instances you want to measure.

   The schematic opens in a new tab. Use the *Escape* key to end selection.

   **Note:** The schematic must be open before you can select any instances. If you do not select a valid instance, a warning stating that the object selected is not a valid selection object is displayed in the CIW.

5. Click *Options* to set the Spectre options controlling sensitivity analysis.

   The *Sensitivity Options* form appears.




   For more information about the options in the form, see the *Other Simulation Topics* chapter of the *Spectre Circuit Simulator Reference*.

6. Optionally, choose *Simulation – Options – Analog* to open the *Simulator Options* form.


© 2023 Cadence Design Systems, Inc.

In the *Sensitivity Options* section on the  tab, specify a filename for the Spectre sensitivity results in the *sensfile* field.

This file is in ASCII format, and is generated in the psf directory. If you do not specify a value, the file is named `sens.output` by default.

**7.** Choose *Results – Print – Sensitivities* to view your results.

The results are displayed in a print window.


© 2023 Cadence Design Systems, Inc.

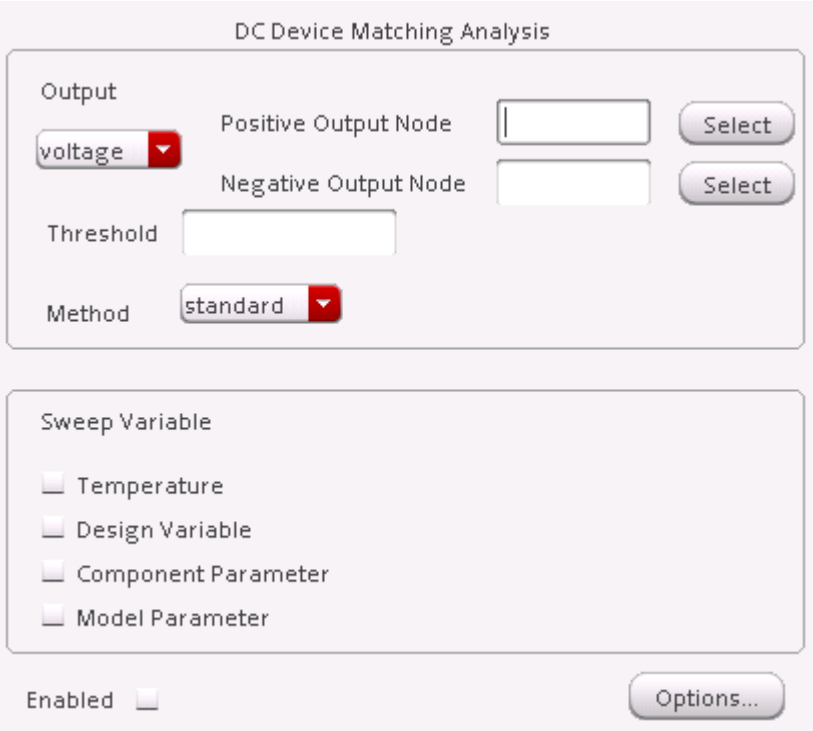
## DC Match Analysis

The *dcmatch* analysis option performs DC device mis-matching analysis for a given output. It computes the deviation in the DC operating point of the circuit caused by mismatch in the devices. Users need to specify mismatch parameters in their model cards for each device contributing to the deviation. The analysis uses the device mismatch models to construct equivalent mismatch current sources to all the devices that have mismatch modeled. These current sources will have zero mean and some variance. The variance of the current sources is computed according to mismatch models. The analysis computes the 3-sigma variance of dc voltage or current due to the mismatch current sources.

Perform the following steps to set up a DC mismatch analysis for the Spectre simulator:

1. In the *Choosing Analyses* form, choose *dcmatch*.

   The *Choosing Analyses* form is updated to show the additional fields for DC mismatch analysis.

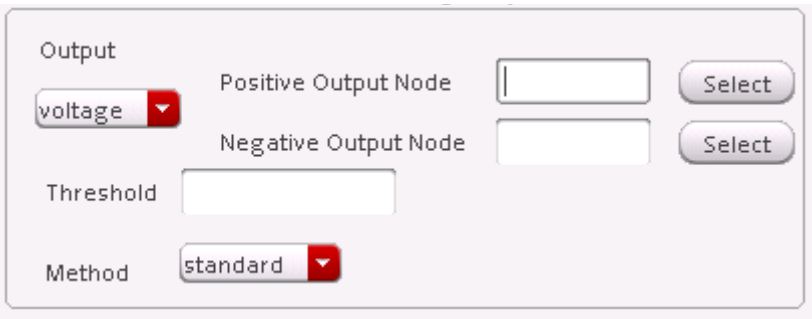



2. Specify the output in the *Output* section of the form. You can choose either *Voltage* or *Probe* in the drop-down list.

   To specify a *Voltage* output,


© 2023 Cadence Design Systems, Inc.

**a.** Choose *Voltage* in the drop-down list.

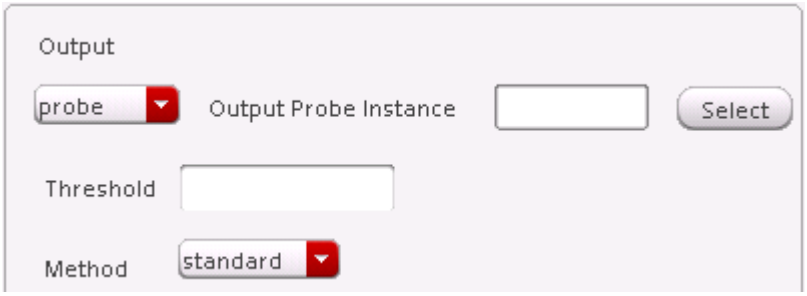



**b.** Click *Select* opposite *Positive Output Node* and click a net in the schematic for the positive output node. Optionally, click *Select* opposite *Negative Output Node* and click a net in the schematic.

To specify a current output,

**a.** Choose *Probe* in the drop-down list.

**b.** Click *Select* opposite O*utput Probe Instance* and click a probe in the schematic.




**Note:** The selected probe device needs to have its terminal currents as network variables. For any other device selection, a warning will be displayed in the CIW, stating that the object selected is not a valid selection object.

Valid Spectre Devices and corresponding analogLib Cells.

| Device | Corresponding analogLib Cells |
|---|---|
| inductor | ind, pinductor |
| vsource | vdc, vpulse, vpwl, vpwlf, vsin, vexp, vsource |
| switch | sp1tswitch, sp2tswitch, sp3tswitch, sp4tswitch |
| tline | tline |


© 2023 Cadence Design Systems, Inc.

| Device | Corresponding analogLib Cells |
|---|---|
| controlled voltage source | vcvs, ccvs, sccvs, svcvs, zccvs, zvcvs, pvcvs, pvcvs2, pvcvs3, pccvs |
| iprobe | iprobe |

If the selected probe has multiple ports (for example tline), you can specify the port number in the *Port* field.

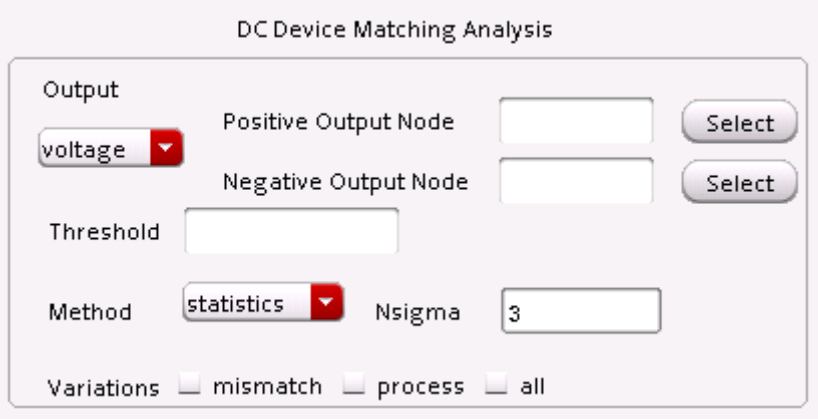
Refer to the Component Description Format User Guide for information on creating more library components selectable for an analysis.

3. Specify a value in the *Threshold* field to control the number of devices displayed in the output log.

   The value should be a positive number, less than or equal to 1. All devices whose relative contribution falls below the specified threshold are not displayed in the output log.

4. In the *Method* drop-down, select `standard` to proceed with standard device mismatch models, or select `statistics` to utilize the parameters defined in statistics blocks to compute output variation

   When the method is set to `statistics`, the following additional fields appear:




- *Nsigma*—Sigma value for the statistical variable. The default value is `3`.
- *Variations*—Type of statistical parameters that are involved in the dcmatch analysis. Possible values are `mismatch`, `process`, or `all`.

| | |
|---|---|
| *Process* | Process statistical variations |
| *Mismatch* | Per-instance statistical variations |


© 2023 Cadence Design Systems, Inc.

*All* Both process and per-instance statistical variations

5. Choose a parameter to sweep in the analysis. The parameters that you can select are *Temperature*, *Design Variable*, *Component Parameter* and *Model Parameter*.

   When you select any of these parameters, the *Sweep Range* section is displayed. Also, the form is updated according to the parameter that is selected.

   **Note:** The *Sweep Variables* section is disabled when method is set to `statistics`.

6. Specify the sweep range and sweep type for the swept parameter.

   Enter the start and stop points of the range or the center and span of the range.

   The sweep type options are mapped to Spectre statements:

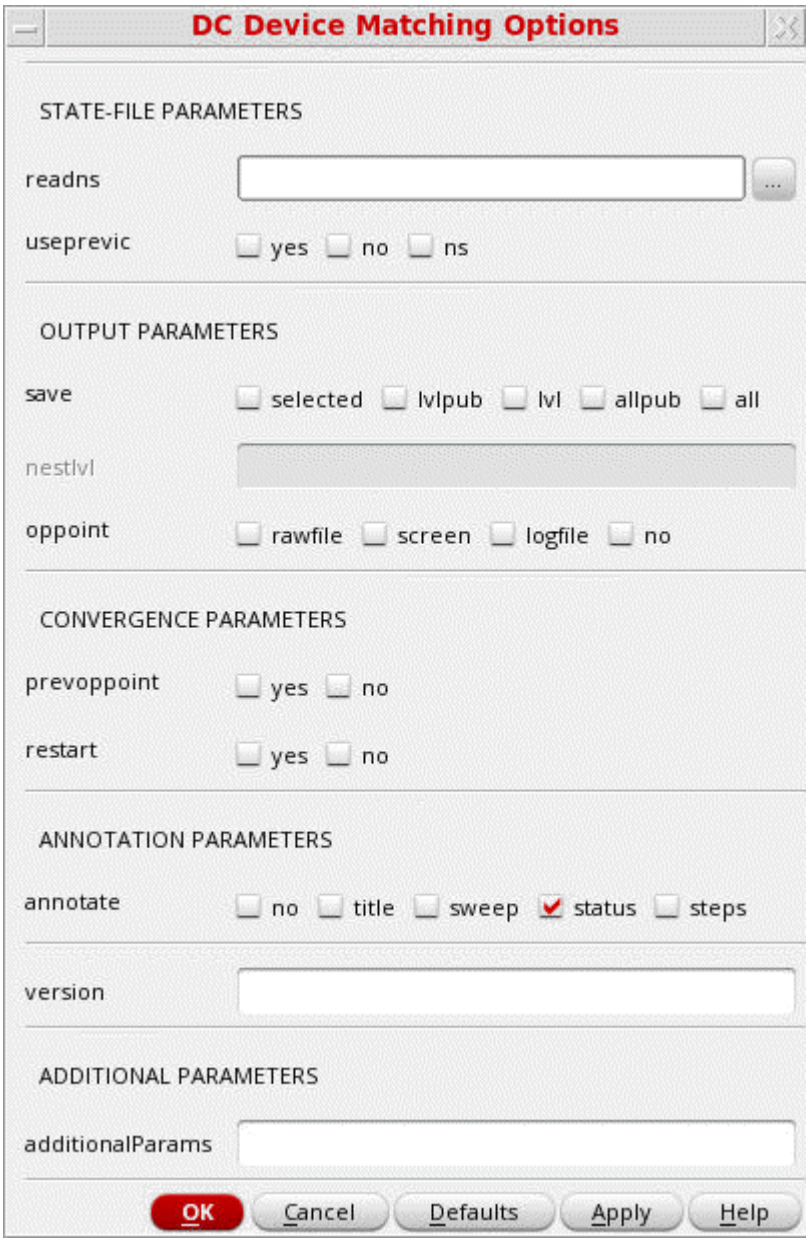
   - Linear + Step Size = step
   - Linear + Number of Steps = lin
   - Logarithmic + Points Per Decade = dec
   - Logarithmic + Number of Steps = log
   - Add Specific Points = values=[…]
   - Add Points By File = valuesfile

   **Note:** the *Sweep Range* section is not displayed when no sweep variable is selected.

7. Click *Options* to open the *Options* form corresponding to the dcmatch analysis.


© 2023 Cadence Design Systems, Inc.

The *DC Device Matching Options* form appears.




For more details about the DC Match analysis, Refer to the *Analysis Statements* chapter in the *Spectre Circuit Simulator Reference* for details.

8. Select the *Enabled* check box and click *Apply*.

To access the results post simulation, choose *Results* – *Print* – *Mismatch Summary*.

**Note:** To print these results from OCEAN, use the OCEAN command, `dcmatchSummary`.


© 2023 Cadence Design Systems, Inc.

## AC Match Analysis

The *acmatch* analysis option is used to linearize the circuit around the DC operating point and computes the variations of AC responses due to statistical parameters defined in statistical blocks. This analysis considers only the mismatch parameters and skips the process parameters.

AC match takes each parameter defined in the statistics blocks and applies variations one at a time, to compute the sensitivity of the output signal with respect to the statistical parameter. Based on the sensitivities computed by this procedure, the total variation of the output is computed by adding the variation contributions from each statistical parameter, assuming that the parameters are mutually independent. The output result is sorted based on the real part of the output sigma of each parameter (or instance).

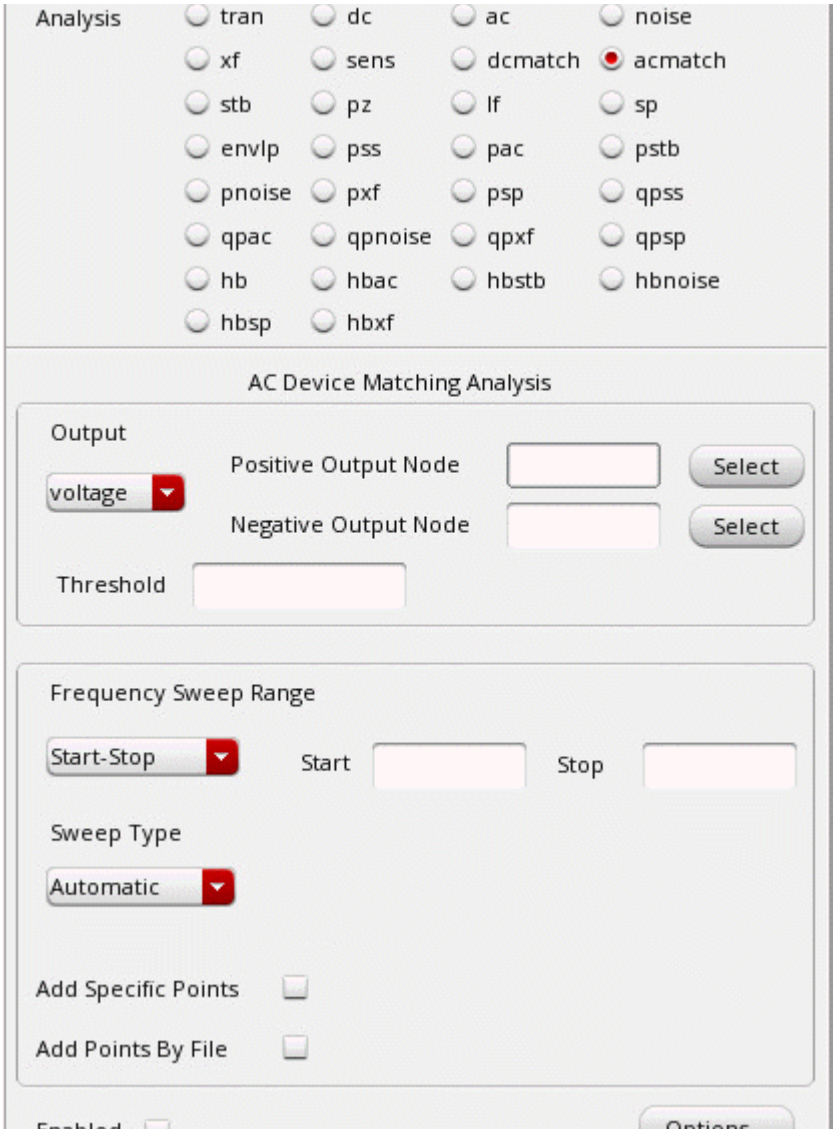
You can specify two or less than two nodes with the AC match analysis. If you specify one node, the analysis outputs the AC response on that node. If you specify two nodes, the analysis outputs the difference in AC response between the two nodes.

Perform the following step to set up a AC match analysis for the Spectre simulator:

1. In the *Choosing Analyses* form appears, choose *acmatch*.


© 2023 Cadence Design Systems, Inc.

The *Choosing Analyses* form is updated to show the additional fields for AC match analysis.

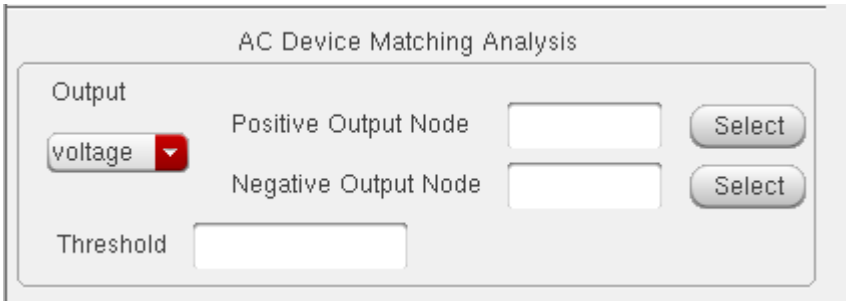



In the *AC Device Matching Analysis* section, specify the following fields:

- *Output* — Specify the output type from the drop-down list. The available values are: `voltage` and `probe`. Default value: `voltage`.

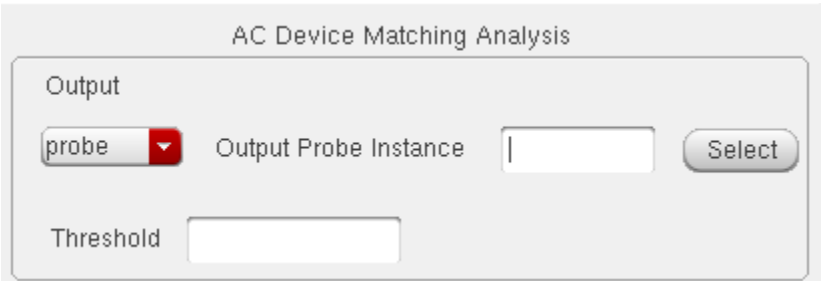

© 2023 Cadence Design Systems, Inc.

To specify a voltage output, select output as `voltage`. The following fields are displayed:




- *Positive Output Node*—Type the name of the positive output net, or click the *Select* to specify a net from the schematic.
- *Negative Output Node*—Type the name of the negative output net, or click the *Select* to specify a net from the schematic.
- *Threshold*—Specify the threshold value used to control the number of devices displayed in the output log. The value should be a positive number less than or equal to 1. All devices whose relative contribution fall below the specified threshold are not displayed in the output log.

To specify a current output, select the output as `probe`. The following fields are displayed:




- *Output Probe Instance*—Type the name of the probe instance, or click the *Select* to specify the probe from the schematic.

  **Note:** The selected probe instance should have its terminal currents set as network variables. If you select any other device, a warning is displayed in CIW stating that the selected instance is not a valid selection object.

The table below lists the valid Spectre devices and its corresponding analogLib Cells:

| Device | Corresponding analogLib Cells |
|---|---|
| inductor | ind, pinductor |


© 2023 Cadence Design Systems, Inc.

| Device | Corresponding analogLib Cells |
|---|---|
| vsource | vdc, vpulse, vpwl, vpwlf, vsin, vexp, vsource |
| switch | sp1tswitch, sp2tswitch, sp3tswitch, sp4tswitch |
| tline | tline |
| controlled voltage source | vcvs, ccvs, sccvs, svcvs, zccvs, zvcvs, pvcvs, pvcvs2, pvcvs3, pccvs |
| iprobe | iprobe |

**Note:** If the selected probe has multiple ports (for example `tline`), you can specify the port number in the *Port* field.

- In the *Frequency Sweep Range* section, specify the following fields:

In the drop-down list, select the method to select the sweep range and type for the parameter sweep. The available methods are: `Start-Stop`, `Center-Span`, and `Single-Point`. The default method is `Start-Stop`.

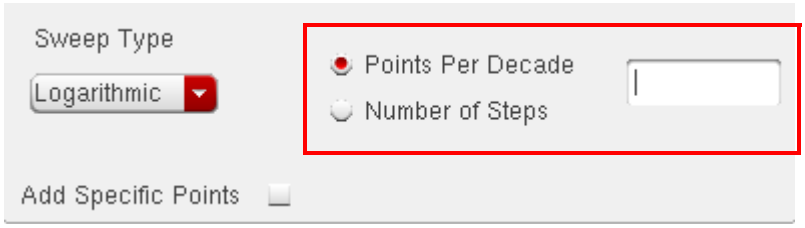
- When you select `Start-Stop`, the *Start* and *Stop* fields appear in which you specify the start and stop values.
- In the *Sweep Type* drop-down, select the sweep type method you want to apply. The available types are: `Automatic`, `Linear` and `Logarithmic`. The default method is `Automatic`.


© 2023 Cadence Design Systems, Inc.

- When you select `linear`, two additional fields appear that you can use to specify the step value:

  You can either select *Step Size* and provide the step size value, or select the *Number of Steps* and provide the value in the text box.

- When you select `Logarithmic`, the following two additional fields appear that you can use to specify the step value:




  You can either select *Points Per Decade* and provide the value, or select the *Number of Steps* and provide the value in the text box.

- Select the *Add Specific Point* check box to specify a particular sweep point.

**Note:** The sweep type options are mapped to Spectre statements:

- Linear + Step Size = step
- Linear + Number of Steps = lin
- Logarithmic + Points Per Decade = dec
- Logarithmic + Number of Steps = log
- Add Specific Points = values=[…]
- Add Points By File = valuesfile

- Click the *Enabled* check box at the end of the form to enable the AC match analysis. By default, this button is not selected.
- Click the *Options* button to specify the options for AC device matching. For more information, see Specifying Options for AC Device Matching.

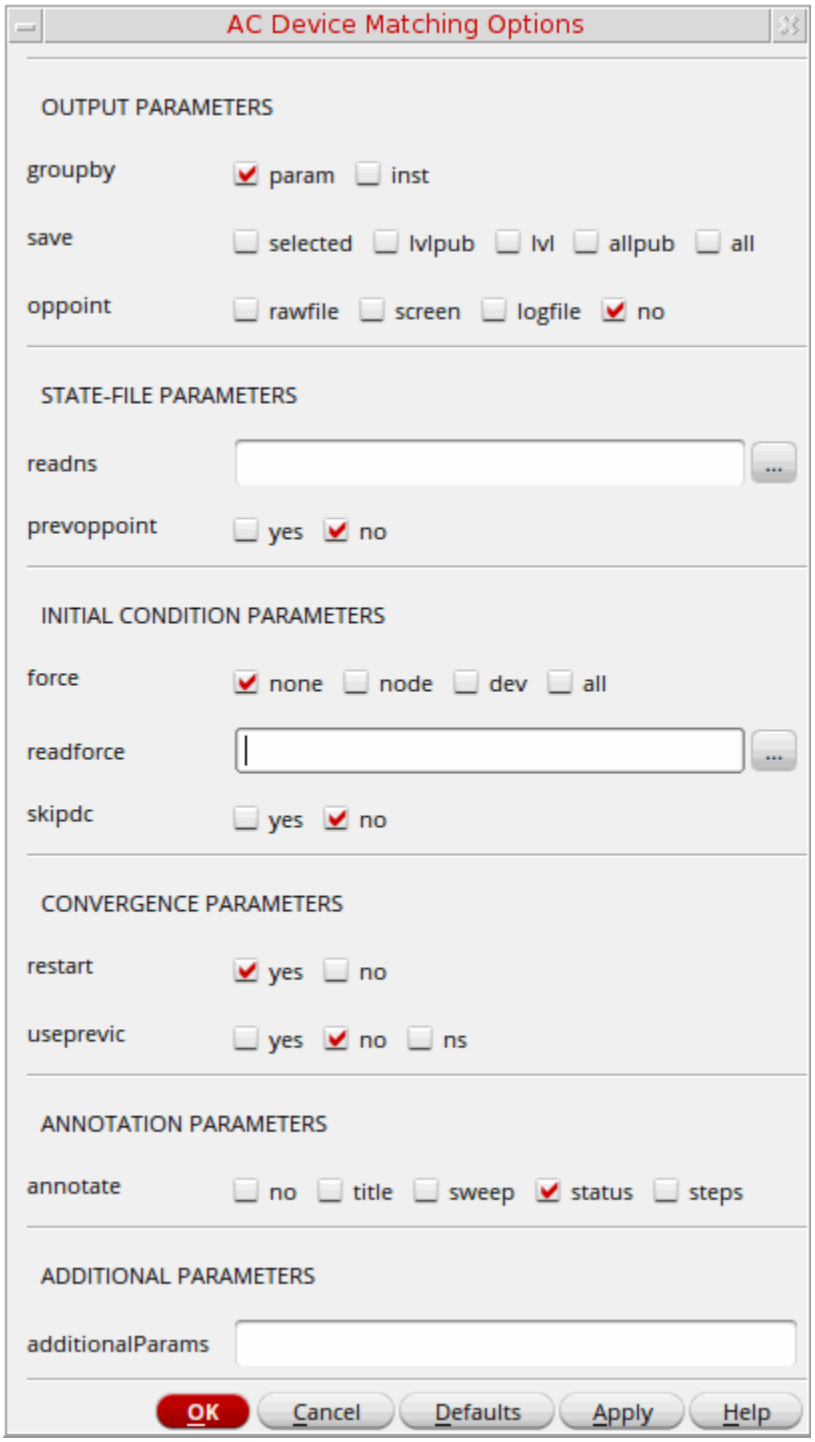

© 2023 Cadence Design Systems, Inc.

## Specifying Options for AC Device Matching

To specify the options used for AC device matching analysis, select the analysis as *acmatch* and click *Options* in the *Choosing Analysis* form.


© 2023 Cadence Design Systems, Inc.

The *AC Device Matching Options* form appears.




This form includes the following fields:


© 2023 Cadence Design Systems, Inc.

*OUTPUT PARAMETERS*

- *groupby*—Specify whether you want to create a group on the basis of parameters or instances.
- *save*—Specify the signals that you want to save in the output. Possible values are `selected`, `lvlpub`, `lvl`, `allpub`, and `all`.
- *oppoint*—Determines whether operating point information should be computed. If `yes`, determine where it should be printed (screen or file). Operating point information is not printed if the operating point computed in the previous analysis remains unchanged. Possible values are `no`, `screen`, `logfile`, and `rawfile`. Default value: `no`.

*STATE-FILE PARAMETERS*

- *readn*—Specify the file that contains an estimate of DC solution (nodeset) by typing its filename. Alternatively, click the ellipses button to select the file.
- *prevoppoint*—Determine whether to use the operating point computed on the previous analysis. Possible values are `no` and `yes`. Default value: `no`.

*INITIAL CONDITION PARAMETERS*

- *force*—Force certain circuit variables to use the values given in the `ic` file, `ic` statements, or `ic` parameter on the capacitors and inductors during the initial operating point AC analysis. The `ic` parameter controls the interaction of various methods of setting the force values. Possible values are:
  - `none`—All initial conditions are ignored.
  - `node`—The ic statements are used, and the ic parameters on the capacitors and inductors are ignored.
  - `dev`—The ic parameters on the capacitors and inductors are used, and the ic statements are ignored.
  - `all`—Both ic statements and ic parameters are used, with the ic parameters overriding the ic statements.

  Default value: `none`.
- *readforce*—Specify the file that contains force values by typing its filename. Alternatively, click the ellipses button to select the file.
- *skipdc*—If set to `yes`, there is no DC analysis for the initial transient. Possible values are `yes` and `no`. Default value: `yes`.

*CONVERGENCE PARAMETERS*


© 2023 Cadence Design Systems, Inc.

- *restart*—Restart the DC solution from scratch if any condition has changed. If the condition is not changed, use the previous solution as the initial guess. Possible values are `no` and `yes`. Default value: `yes`.
- *useprevic*—If set to yes or `ns`, use the converged initial condition from previous analysis as ic or ns. Possible values are `no`, `yes`, and `ns`. Default value: `no`.

*ANNOTATION PARAMETERS*

- *annotate*—Degree of annotation. Possible values are `no`, `title`, `sweep`, `status`, `steps`. Default value: `step`.

*ADDITIONAL PARAMETERS*

- *additionalParams*—Specify additional parameters.

For more information about these options, refer to the *Analysis Statements* chapter in the *Spectre Circuit Simulator Reference*.


© 2023 Cadence Design Systems, Inc.

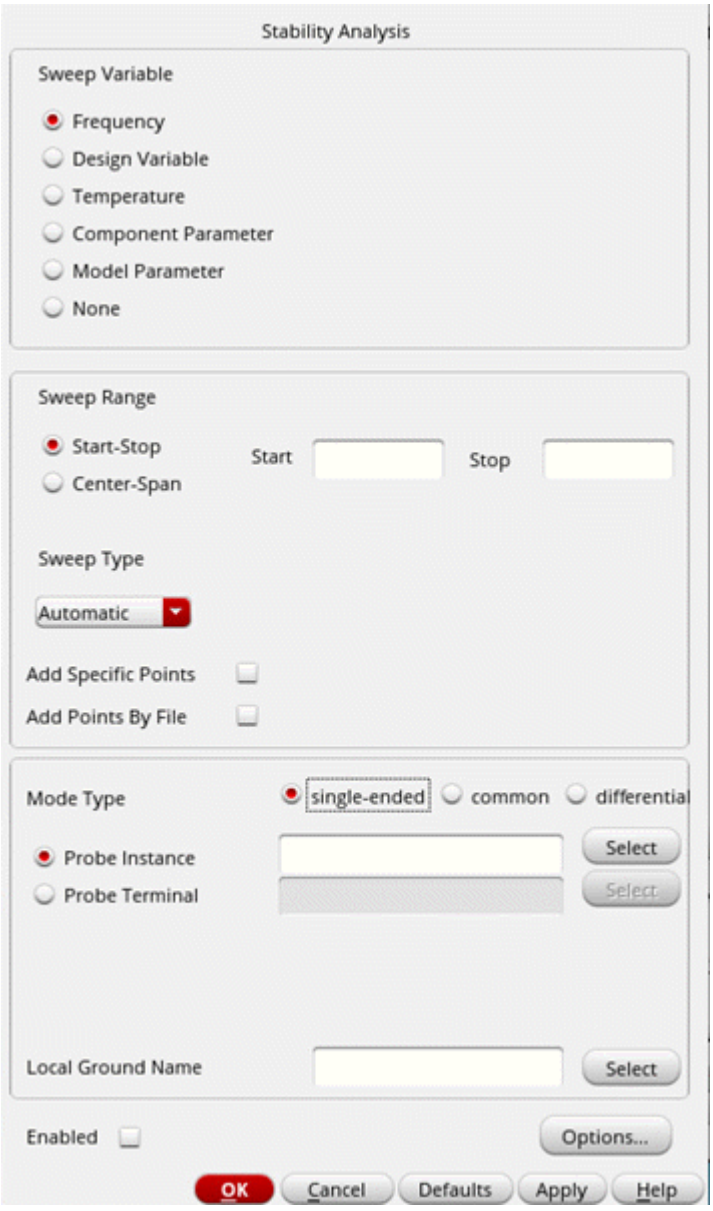
## Stability Analysis

Stability analysis outputs the loop gain for the feedback loop or a gain device.

Perform the following steps to set up a stability analysis for the Spectre simulator:

1. In the *Choosing Analyses* form, choose *stb*.

   The *Choosing Analysis* form is updated to show the additional fields for stability analysis.





© 2023 Cadence Design Systems, Inc.

2. Choose a parameter to sweep in the analysis. The parameters that you can select are *Frequency*, *Design Variable*, *Temperature*, *Component Parameter* and *Model Parameter*.

   - For any parameter other than frequency, you need to specify the frequency at which the analysis is to be performed.
   - When the swept parameter is frequency, it also outputs the phase and gain margins if they can be calculated from the loop gain curve within the swept frequency values.

3. Specify the sweep range and sweep type for the swept parameter.

   Enter the start and stop points of the range or the center and span of the range.

   The sweep type options are mapped to Spectre statements:

   - Linear + Step Size = step
   - Linear + Number of Steps = lin
   - Logarithmic + Points Per Decade = dec
   - Logarithmic + Number of Steps = log
   - Add Specific Points = values=[…]
   - Add Points By File = valuesfile

   The form changes dynamically as per the current selection.

4. From the *Mode Type* options, select one of the following types of circuit for which you want to specify the loop opening points to calculate the loop gain.

   - *single-ended*: Select this for single-ended circuits.
   - *common*: Select this for circuits operating in the common mode.
   - *differential*: Select this for differential circuits.

   Based on the selected method, the form changes dynamically to display the applicable options. The loop gain is then calculated according to the probe instances or terminals selected through these options.

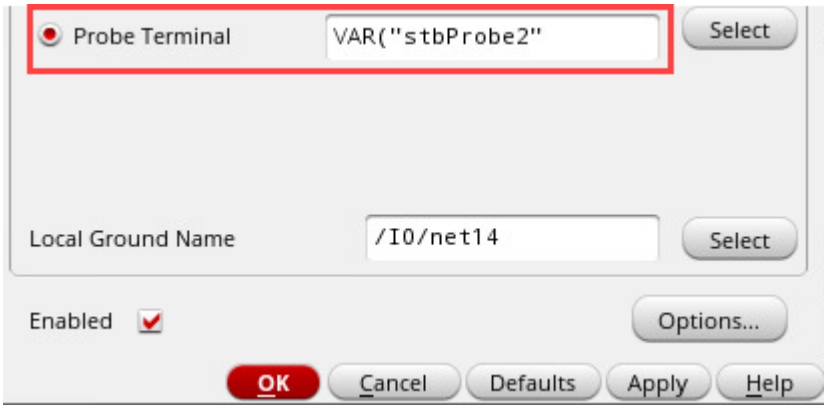
   - For *single-ended* mode, select one of the following options:
     - *Probe Instance*: Use the *Select* button to select the probe instance from the schematic. The name of the selected instance automatically appears in the text field.

       For single-ended mode, specify an iprobe or a zero-volt source. For common or differential mode, specify a diffstbprobe.


© 2023 Cadence Design Systems, Inc.

- *Probe Terminal*: Use the *Select* button to select a probe terminal from the schematic and specify a leaf terminal. The name of the selected terminal automatically appears in the text field.

  Alternatively, specify the value through the VAR function.

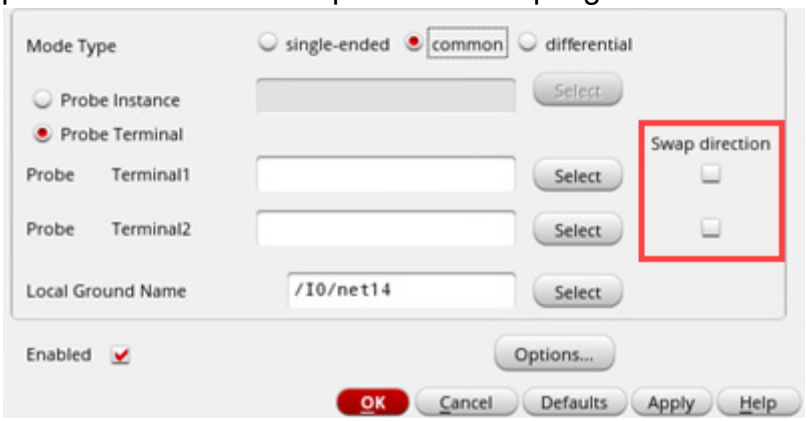



- For *common* mode, select one of the following options.

  - *Probe Instance*: Use the *Select* button to select the probe instance from the schematic. The name of the selected instance automatically appears in the text field.

  - *Probe Terminal*: Use the *Select* button to select the probe terminals for the fields *Probe Terminal1* and *Probe Terminal2* from the schematic. The names of the selected terminals automatically appear in the text fields.

    (Optional) In the *Swap direction* section, select the check box next to the field specifying the probe instance or terminal for which you want to change the probe direction with respect to the loop signal flow.

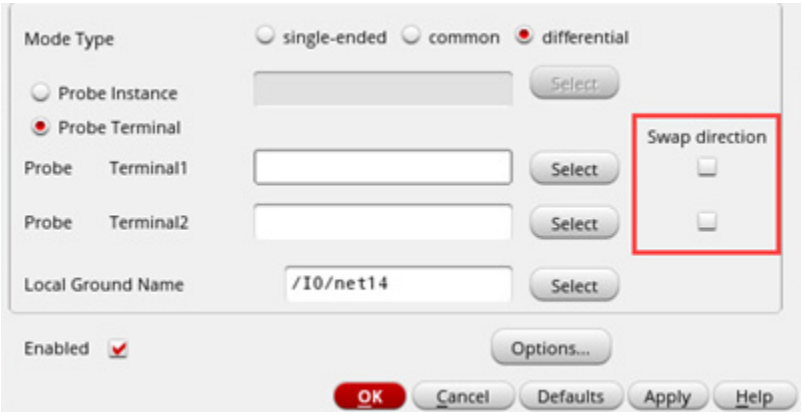



- For *differential* mode, select one of the following options.


© 2023 Cadence Design Systems, Inc.

- *Probe Instance*: Use the *Select* button to select the probe instance from the schematic. The name of the selected instance automatically appears in the text field.
- *Probe Terminal*: Use the *Select* button to select the probe terminals for the fields *Probe Terminal1* and *Probe Terminal2* from the schematic. The names of the selected terminals automatically appear in the text fields.

  (Optional) In the *Swap direction* section, select the check box next to the field specifying the probe instance or terminal for which you want to change the probe direction with respect to the loop signal flow.

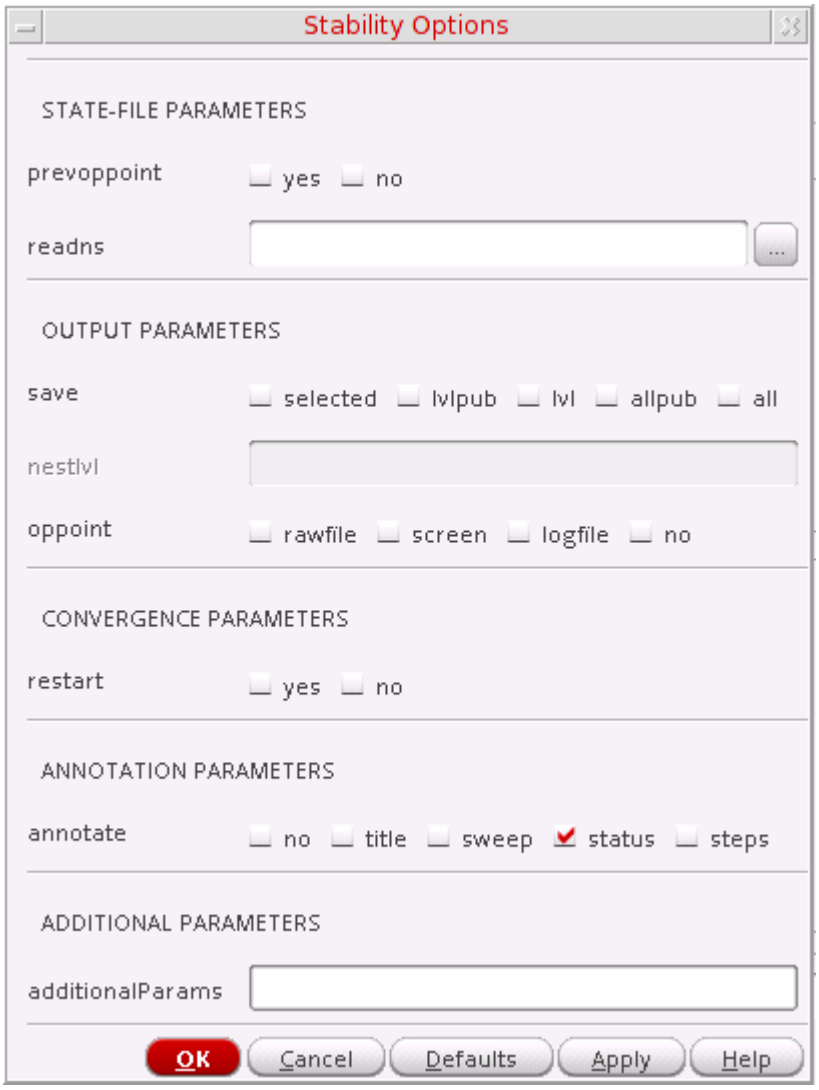



5. Specify a value in the *Local Ground Name* text field. You can use the *Select* button to select the node from the schematic and the name for the selected node automatically appears in the text field.


© 2023 Cadence Design Systems, Inc.

The *Stability Options* form appears.

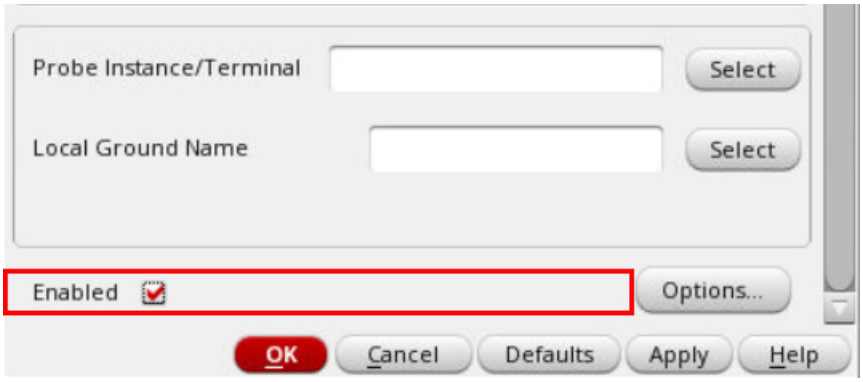
.




To know more on setting up a stability analysis, refer to the *Analysis Statements* chapter in the *Spectre Circuit Simulator Reference* for details.


© 2023 Cadence Design Systems, Inc.

6. After setting up the options, select the *Enabled* check box and click *Apply*.




To access the results post simulation, choose *Results – Print – Stability Summary*.

You can also access these results from *Results – Direct Plot.*


© 2023 Cadence Design Systems, Inc.

## Pole-Zero Analysis

Pole-Zero analysis is a useful method for studying the behavior of linear time invariant networks and can be applied to the design of analog circuits. Therefore, it can be used for determining stability of designs.

In pole-zero analysis, a network is described by its network transfer function. For any linear time invariant network, it can be written in the general form:

$$H(S) = \frac{N(S)}{D(S)} = \frac{a_0 S^m + a_1 S^{m-1} + \ldots + a_m}{b_0 S^n + b_1 S^{n-1} + \ldots + b_n}$$

Similarly, in the factorized form:

$$H(S) = \frac{N(S)}{D(S)} = \frac{a_0}{b_0} \cdot \frac{(S+Z_1)(S+Z_2)}{(S+P_1)(S+P_2)} \cdots \frac{(S+Z_i)}{(S+P_i)} \cdots \frac{(S+Z_m)}{(S+P_m)}$$

Here, the roots of the numerator **`N(S)`** (that is, **`Z`**) are called *zeros* of the network function. The roots of the denominator **`D(S)`** (that is, ***P***) are called the *poles* of the network function. ***S*** is the complex frequency.

The behavior of the network depends upon the location of the poles and zeros on the complex S-plane. The poles are called natural frequencies of the network.

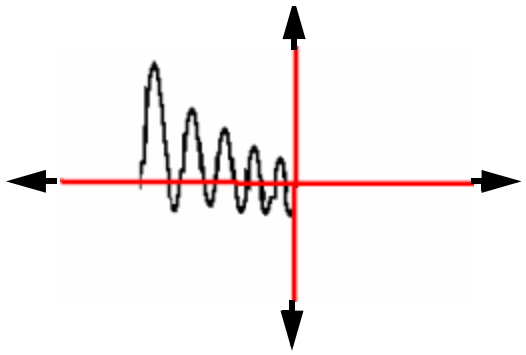
For example:

$$H(S) = \frac{(S-2) \cdot (S+1)}{S}$$

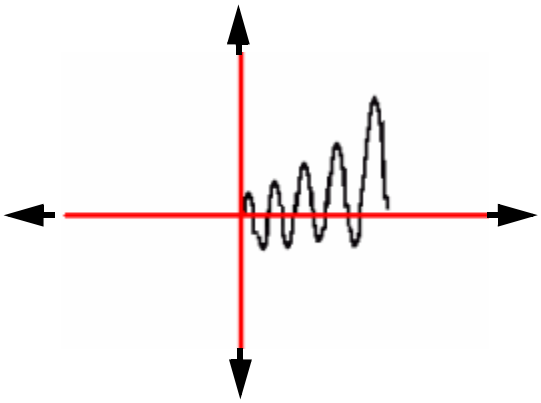
Here, the *zeros* are the values of ***H(S)*** which make it zero (***S***=`2` and ***S***=`-1`). The *poles* make ***H(S)*** go to infinity (the pole is at ***S***=`0`)


© 2023 Cadence Design Systems, Inc.

When all the poles have negative real parts, the poles are located on the left hand side of the XY plane. In this situation, the circuit is considered *stable*. The following diagram illustrates the behavior of a stable circuit:



In case there are poles present on the right hand side of the XY plane, the circuit is considered *unstable*. The following diagram illustrates the behavior of an unstable circuit.



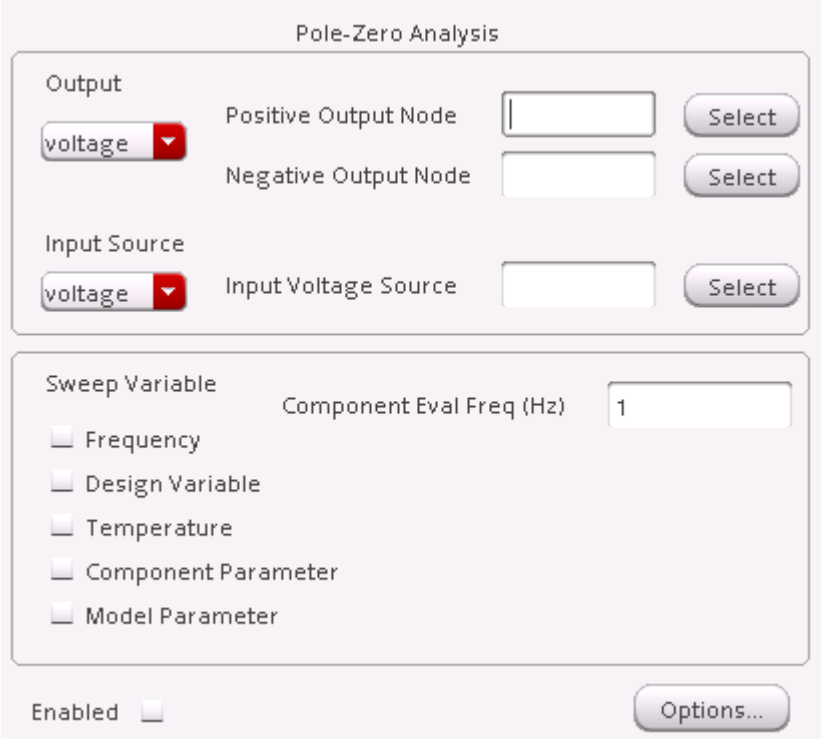
For absolute stability, there can be no poles with positive real parts. If there are poles with positive real parts the output signal may become unbounded.

Perform the following steps to set up a pole-zero analysis for the Spectre simulator:

1. In the *Choosing Analyses* form, choose *pz.*


© 2023 Cadence Design Systems, Inc.

The *Choosing Analyses* form is updated to show the additional fields for pole-zero analysis.




2. Specify the output in the *Output* section of the form. You can choose either *Voltage* or *Probe* in the cyclic drop down field.

   To specify a *Voltage* output,

   a. Choose *Voltage* in the drop-down list.

   b. Click *Select* opposite *Positive Output Node* and click a net in the schematic for the positive output node.

      Also, click *Select* opposite *Negative Output Node* and click a net in the schematic.

   To specify a current output,

   a. Choose *Probe* in the drop-down list.

   b. Click *Select* opposite O*utput Probe Instance* and click an instance, with terminal currents as network variables, in the schematic.

   For any other device selection, a warning will be displayed in the CIW stating that the object selected is not a valid selection object.


© 2023 Cadence Design Systems, Inc.

Valid Spectre Devices and corresponding analogLib cells:

| Device | Corresponding analogLib Cells |
|---|---|
| inductor | ind, pinductor |
| vsource | vdc, vpulse, vpwl, vpwlf, vsin, vexp, vsource |
| switch | sp1tswitch, sp2tswitch, sp3tswitch, sp4tswitch |
| tline | tline |
| controlled voltage source | vcvs, ccvs, sccvs, svcvs, zccvs, zvcvs, pvcvs, pvcvs2, pvcvs3, pccvs |
| iprobe | iprobe |

When *tline* is the device selected, the *Output* section of the form is updated to show the *porti* field. This parameter helps you specify a current output that is defined by the device terminal current. Since all of these are two-terminal devices, the current through one of the device terminals would be the same as through the other. The *tline* device is the only one that has more than two terminals.

3. Specify the input voltage or current source by selecting either *voltage* or *current* in the *Input Source* drop-down list of the same form.
4. If you want to sweep a variable in conjunction with the Pole-Zero analysis, choose a parameter to sweep. The parameters that you can select are *Frequency, Design Variable*, *Temperature, Component Parameter* and *Model Parameter*.

   When you select any of these parameters, the form is updated according to the parameter that is selected.
5. Specify the sweep range and sweep type for the swept parameter (*Design Variable*, *Temperature, Component Parameter* or *Model Parameter*).

   Enter the start and stop points of the range or the center and span of the range.

   The sweep type options are mapped to Spectre statements:

   - Linear + Step Size = step
   - Linear + Number of Steps = lin
   - Logarithmic + Points Per Decade = dec
   - Logarithmic + Number of Steps = log

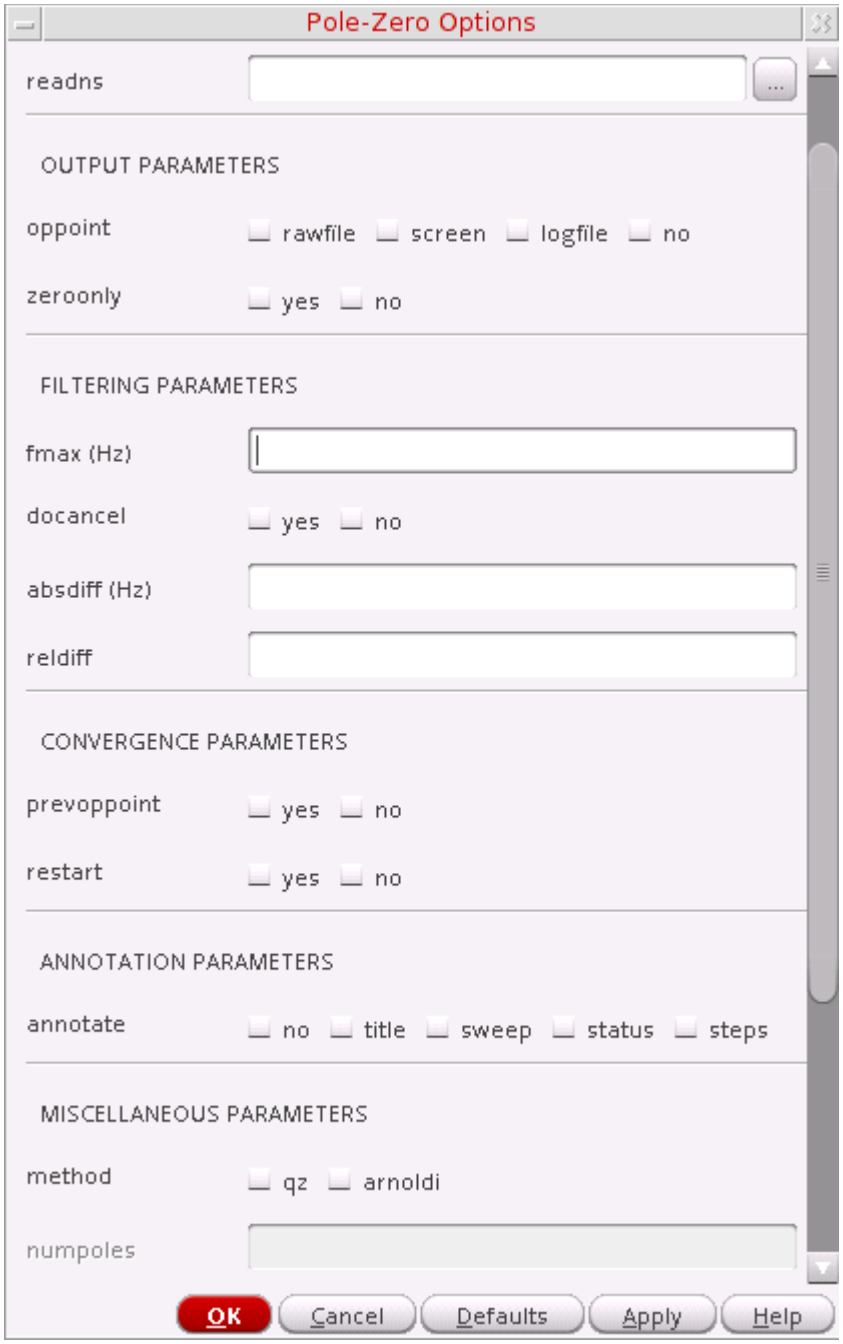

© 2023 Cadence Design Systems, Inc.

- Add Specific Points = values=[…]
- Add Points By File = valuesfile

**Note:** The *Sweep Range* section is not displayed when no sweep variable is selected.


© 2023 Cadence Design Systems, Inc.

6. Click *Options* to open the *Options* form corresponding to the pz analysis. The *Pole-Zero Options* form appears.




For details, refer to the *Analysis Statements* chapter in the *Spectre Circuit Simulator Reference*.


© 2023 Cadence Design Systems, Inc.

*Tip*

You can also type `spectre -h pz` in the shell, for help on Pole-Zero options.

7. Select the *Enabled* check box and click *Apply*.

To print the results post simulation, choose *Results* – *Print* – *Pole-Zero Summary.*

You can also plot results from *Results* – *Direct Plot* – *Main Form.*

To plot and print these results from OCEAN, use the OCEAN command, `pzPlot` and `pzSummary`.


© 2023 Cadence Design Systems, Inc.

## Loop Finder Analysis

The loop finder analysis linearizes the circuit about the DC operating point and identifies the loops that may potentially cause stability problems. The analysis does not require the identification of loops or the use of probes.

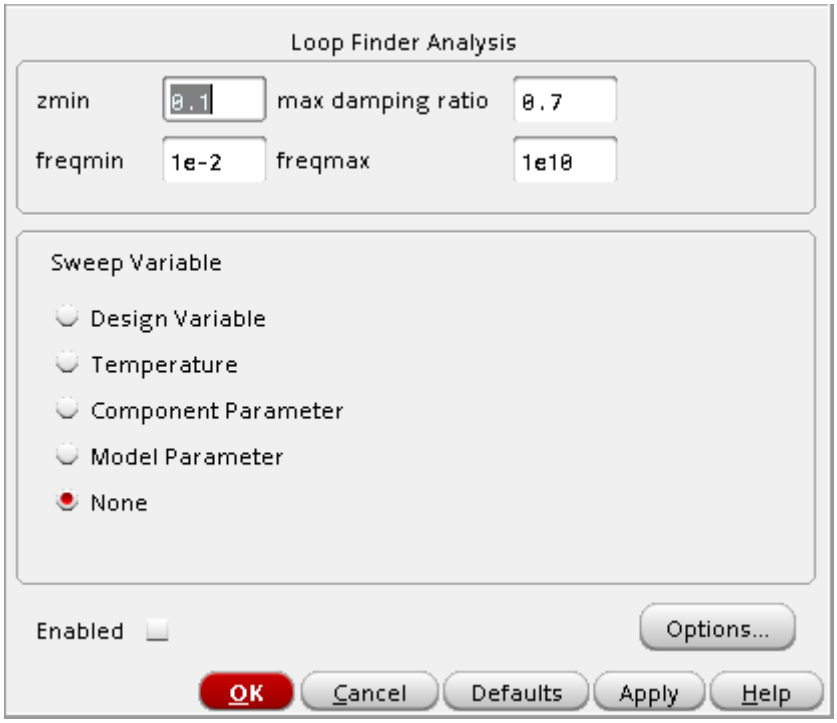
The analysis computes the impedance at a frequency that is equal to the magnitude of the pole and identifies the high-impedance nodes. It displays the impedance of all the high-impedance nodes within a loop, sorted by the impedance value.

The loop finder analysis, by default, uses the dense method to detect the suspected poles. However, this method is computationally intensive and works best on small-to-medium sized circuits. You can use the krylov method (iterative sparse solver) on large circuits for a better performance.

Perform the following steps to set up a loop finder analysis:

1. Select *lf* in the Choosing Analyses form.

   The options required for loop finder analysis appear in the form:




2. In the *zmin* field, specify the minimum DC impedance to be used in loop identification. The default value is `0.1Ω`.


© 2023 Cadence Design Systems, Inc.

You can use this field to control the number of nodes in the output. A higher value means that fewer nodes are displayed in the output for a given loop.

3. In the *max damping ratio* field, specify the maximum damping ratio to be used in pole detection. The default value is `0.7`.

   You can use this field to control the filtering of suspected poles. Setting the value of this field to `1` results in the inclusion of all poles.

4. In the *freqmin* and *freqmax* fields, specify the minimum and maximum natural frequency for targeting loops.

5. In the *sensitivity* field, specify the sensitivity of the krylov method for pole detection and the accuracy for impedance computation. Higher values increase the simulation runtime, but reduce the chance of missing a pole and improve the accuracy of impedance values. Possible values are `1`, `2`, `3`, `4`, and `5`.

6. In the *Sweep Variable* section, choose a sweep variable and specify the necessary parameters.

   - If you want to sweep a design variable, select *Design Variable* and enter the name of the design variable. Or click *Select Design Variable* and select a variable from the *Select Design Variable* list box.
   - If you want to sweep circuit temperature, select *Temperature*.
   - If you want to sweep a component parameter, select *Component Parameter* and specify the component name and parameter name. Or click *Select Component* to open the Schematic window and select the component.
   - If you want to sweep a model parameter, select *Model Parameter* and enter the model and parameter names.
   - If you do not want to specify any sweep variable, select *None*.

7. Specify the sweep range by performing one of the following steps:

   - Select the *Start-Stop* option to specify the start sweep limit in the *Start* field and the stop sweep limit in the *Stop* field.
   - Select the *Center-Span* option to specify the center of the sweep in the *Center* field and the sweep limit span limit in the *Span* field.

8. Select the sweep type from the *Sweep Type* drop-down list box and perform one of the following steps:

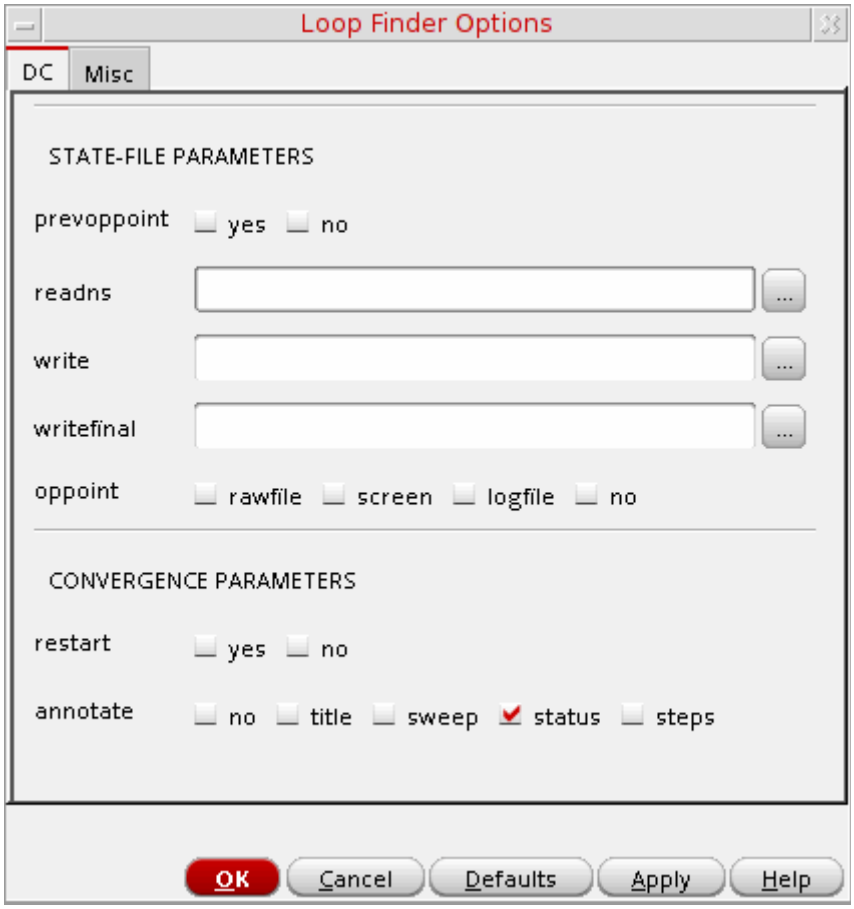
   - If the sweep type is *Linear*, select the *Step Size* option and specify the step size for the linear sweep, or select the *Number of Steps* option and specify the number of steps for the linear sweep.


© 2023 Cadence Design Systems, Inc.

- If the sweep type is *Logarithmic*, select the *Points Per Decade* option and specify the points per decade for the logarithmic sweep, or select the *Number of Steps* option and specify the number of steps for the logarithmic sweep.

9. (Optional) Select the *Add Specific Points* check box to specify a list of values to sweep. Use spaces to separate each value in the field.
10. (Optional) Select the *Add Points By File* check box to choose the file containing the values to sweep. You can either enter the path to the file in the text box manually, or click the *Browse* button next to the text box and choose the file.
11. Click *Options* to set the Spectre options controlling the loop finder simulation.

    The Loop Finder Options form appears:




    For information about the options available in this form, see *Loopfinder Analysis* in the *Spectre Circuit Simulator Reference* manual.

    Click *OK* to close the form.


© 2023 Cadence Design Systems, Inc.

**12.** In the Choosing Analyses form, select the *Enabled* check box and then click *Apply* to save the settings.


© 2023 Cadence Design Systems, Inc.

## Other Spectre Analyses

You can also run the following analysis for Spectre:

- Envelope Following Analysis (envlp)
- Periodic Steady State Analysis (pss)
- Periodic AC Analysis (pac)
- Periodic Stability Analysis (pstb)
- Periodic Noise Analysis (pnoise)
- Periodic XF Analysis (pxf)
- Periodic S-parameter Analysis (psp)
- Quasi-Periodic Steady State Analysis (qpss)
- Quasi-Periodic AC Analysis (qpac)
- Quasi-Periodic Noise Analysis (qpnoise)
- Quasi-Periodic XF Analysis(qpxf)
- Quasi-Periodic S-parameter Analysis (qpsp)
- Harmonic Balance Analysis Analysis (hb)
- Harmonic Balance AC Analysis (hbac)
- Harmonic Balance Noise Analysis (hbnoise)
- Harmonic Balance SP Analysis (hbsp)

For information on the Spectre analyses available, see the *Spectre Circuit Simulator and Accelerated Parallel Simulator RF Analysis User Guide*.

For more help on setting up a particular analysis, refer to the chapter *Analysis Statements* in the *Spectre Circuit Simulator Reference*.

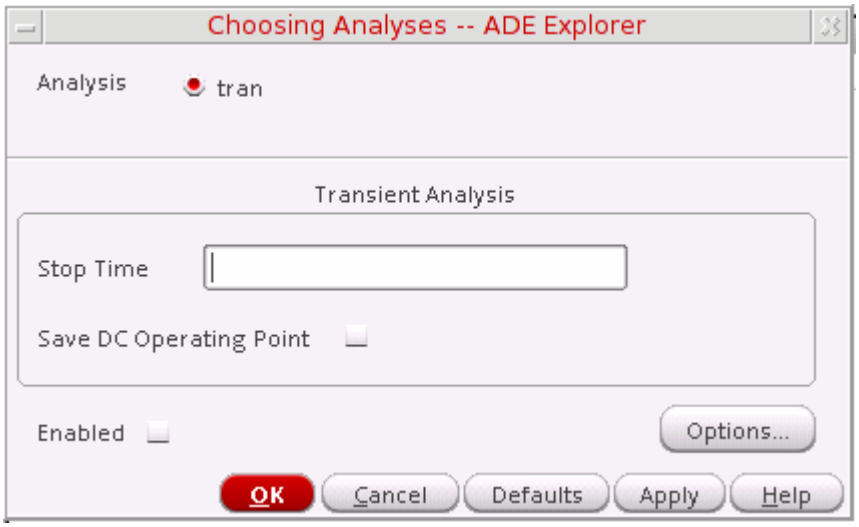

© 2023 Cadence Design Systems, Inc.

# Setting Up an Analysis for UltraSim

Perform the following steps to set up the *Virtuoso® UltraSim™* simulator for analysis:

1. Choose *Analyses – Choose*.

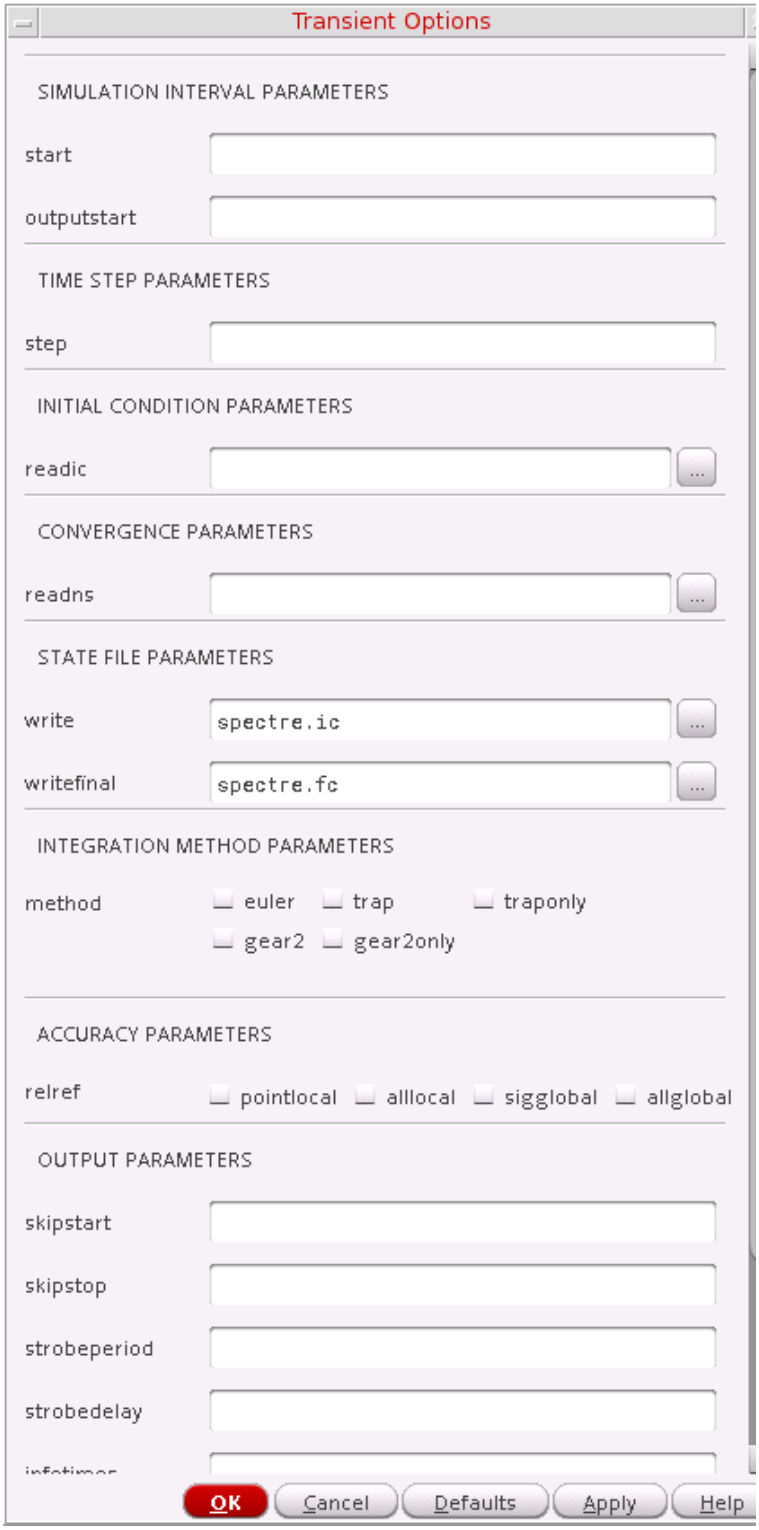
   The *Choosing Analyses* form appears and the *tran* option is selected.




2. Click *Options*.


© 2023 Cadence Design Systems, Inc.

The *Transient Options* form appears.





© 2023 Cadence Design Systems, Inc.

3. Set the transient simulation options as needed.

   - *start=0 s*—Transient start time.
   - *outputstart*—Output is saved after this time is reached.
   - *step* Minimum time step used by the simulator to maintain the aesthetics of the computed waveforms.
   - *readic*—Initial condition is contained in this file (`readic` file is specified relative to the `/netlist` directory).
   - *readns*—Estimate of the DC solution (nodeset) is contained in this file.
   - *write*—Initial transient solution is written to this file.
   - *writefinal*—final transient solution is written to this file.
   - *method*—Integration method. Values are `euler` (default), `trap`, `traponly`, `gear2` or `gear2only`.
   - *relref*—Reference used for the relative convergence criteria. The default is derived from `errpreset`. Possible values are `pointlocal`, `alllocal`, `sigglobal`, and `allglobal`.
   - *skipstart=starttime s*—Time to start skipping output data.
   - *skipstop=stoptime s*—Time to stop skipping output data.
   - *strobeperiod (s)*—Output strobe interval (in seconds of transient time).
   - *strobedelay=0 s*—Delay (phase shift) between the skipstart time and the first strobe point.
   - *infotimes=[...] s*—Times when info analysis specified by `infoname` is performed.
   - *maxstep_window*—Maximum time step (setting `maxstep` to a smaller value can improve simulation accuracy). You can set global or local maxstep option for one instance. However, if you want to add maxstep options for different instances or subckts, you can add a column in maxstep_window and choose string and input in HED. For example, `time1 value1 time2 value 2....`
   - *subckt instance*—Subcircuit blocks for maximum time step.

   **Note:** By default, the output format of the output of an UltraSim transient analysis is SST2 (SignalScan Turbo 2).

   For more information about the transient simulation options, refer to Chapter 2, “Netlist Formats” in the *Virtuoso UltraSim Simulator User Guide*.


© 2023 Cadence Design Systems, Inc.

4. Click *OK*.
5. Choose *Simulation – Netlist and Run*.

   Check the CIW for messages stating that the simulation has started and finished successfully (information is also written to a log file as the simulation runs).


© 2023 Cadence Design Systems, Inc.

# Setting Up an Analysis for AMS

You can run the following analyses for AMS:

- Transient Analysis
- DC Analysis
- AC Analysis
- Noise Analysis
- Stability Analysis
- Transfer Function Analysis

***Related Topics***

Setting Up Analyses


© 2023 Cadence Design Systems, Inc.

# Setting Up an Analysis for HSPICE

You can select the analysis you want to run by using the *Choosing Analyses* form. The analyses that are supported are DC, Transient, AC, Noise, OP, AC Match, DC Match, and LSTB. When you select an analysis, the form displays the fields that are required for the selected analysis.

You can run the following analyses for HSPICE:

- DC Analysis on page 347
- Transient Analysis on page 349
- AC Analysis on page 350
- Noise Analysis on page 351
- OP Analysis on page 352
- AC Match Analysis on page 353
- DC Match Analysis on page 354
- LSTB Analysis on page 355

For information on how to set up an analysis, see Setting Up Analyses on page 246.


© 2023 Cadence Design Systems, Inc.

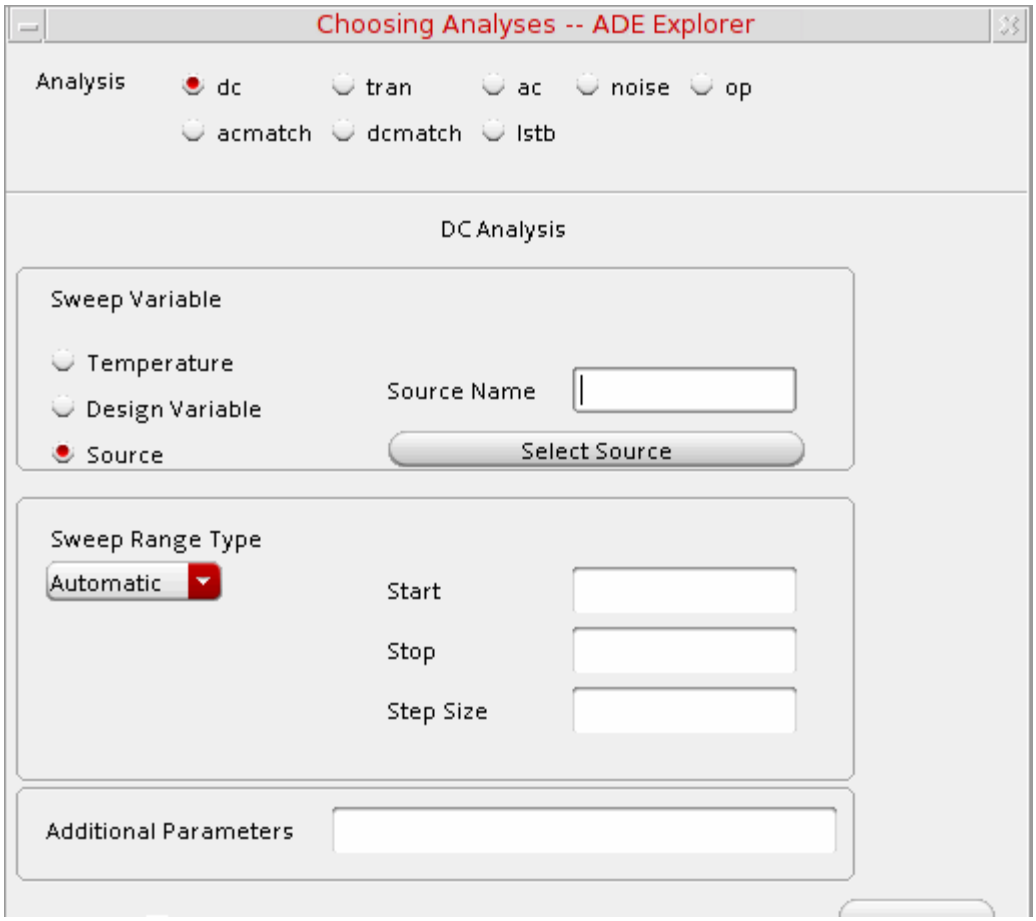
## DC Analysis

DC analysis finds the DC operating point or DC transfer curves of the circuit. To generate transfer curves, specify a sweep variable and a sweep range.

Perform the following step to set up a DC analysis:

1. In the *Choosing Analyses* form, choose *dc*.

   The *Choosing Analyses* form is updated to show the additional fields for DC analysis.

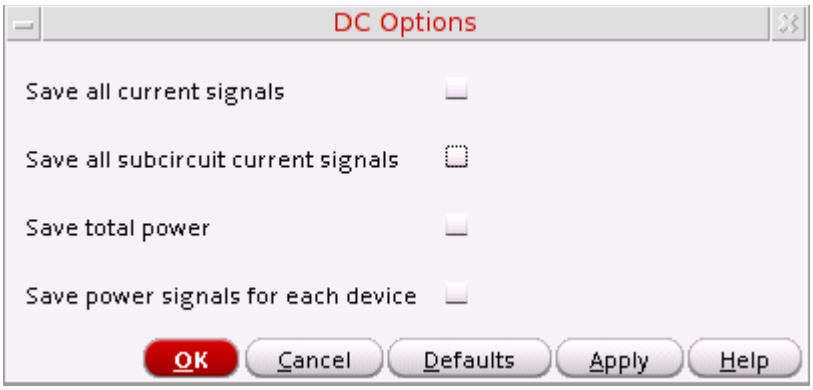



   This form displays the different types of DC sweep variables and sweep range types. For more information about DC Analysis, see DC Analysis on page 278.

2. Click *Options*.


© 2023 Cadence Design Systems, Inc.

The *DC Options* form appears.




3. Set the options as needed and click *Apply*.


© 2023 Cadence Design Systems, Inc.

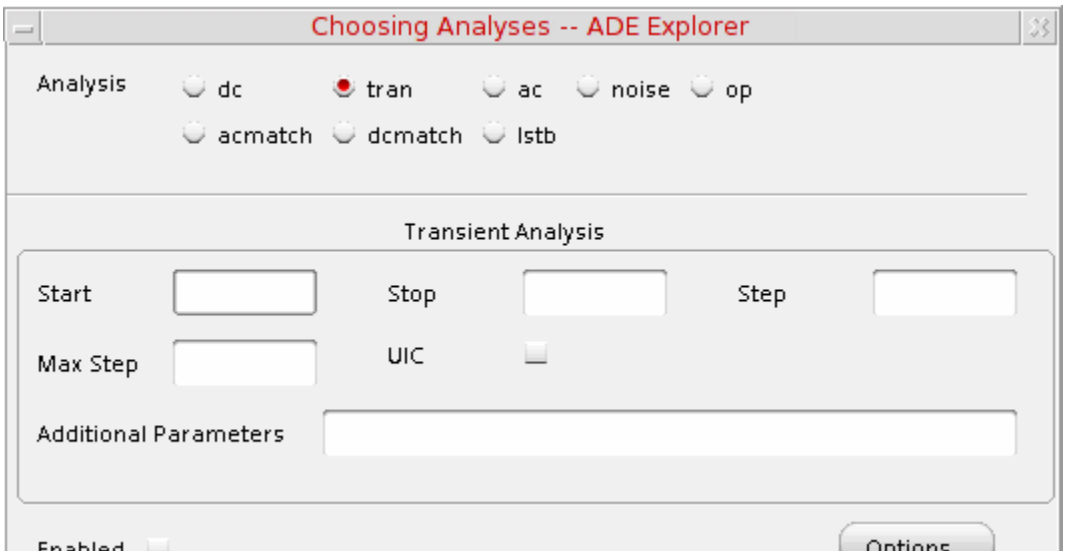
## Transient Analysis

Transient analysis computes the transient response of a circuit over an interval.

Perform the following step to set up a transient analysis:

1. In the *Choosing Analyses* form, choose *tran*.

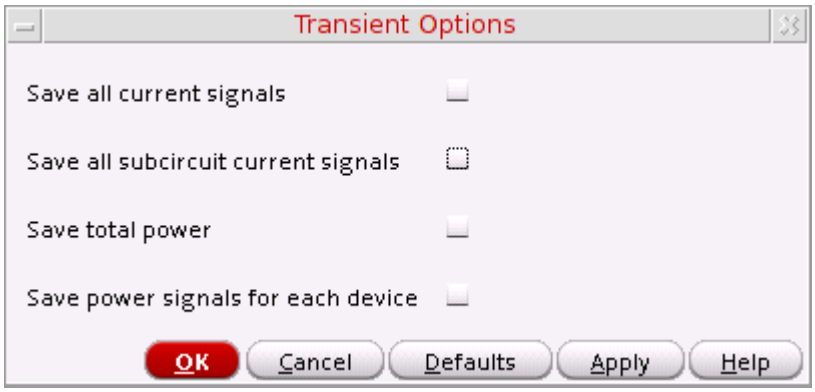
   The *Choosing Analyses* form is updated to show the additional fields for transient analysis.




   For more information about transient analysis, see Transient Analysis on page 250.

2. Click *Options*.

   The *Transient Options* form appears.




3. Set the options as needed and click *Apply*.


© 2023 Cadence Design Systems, Inc.

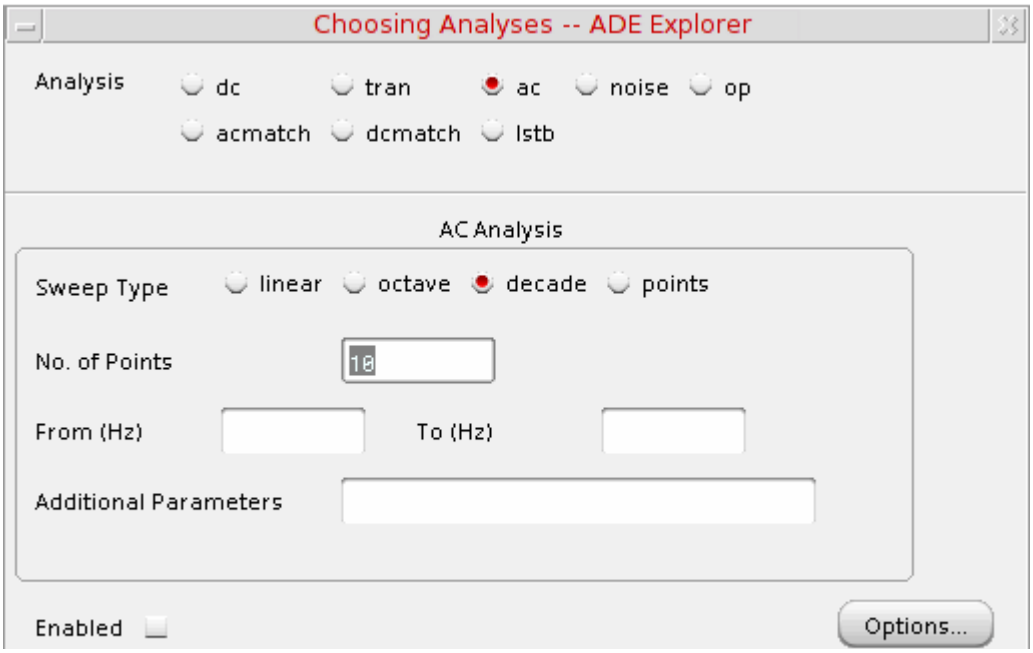
## AC Analysis

AC analysis linearizes the circuit about the DC operating point and computes the response to a given small sinusoidal stimulus.

Perform the following step to set up an AC analysis:

1. In the *Choosing Analyses* form, choose *ac*.

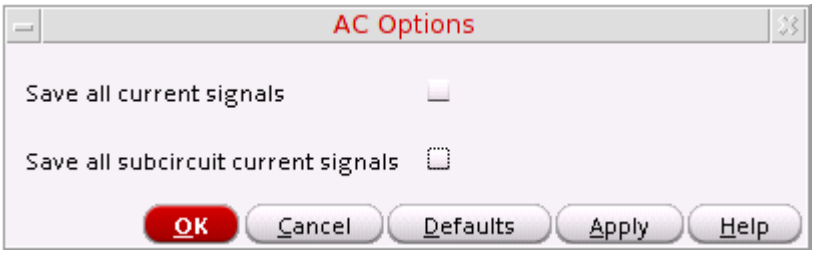
   The *Choosing Analyses* form is updated to show the additional fields for AC analysis.




   For more information about AC analysis, see AC Analysis on page 285.

2. Click *Options*.

   The *AC Options* form appears.




3. Set the options as needed and click *Apply*.


© 2023 Cadence Design Systems, Inc.

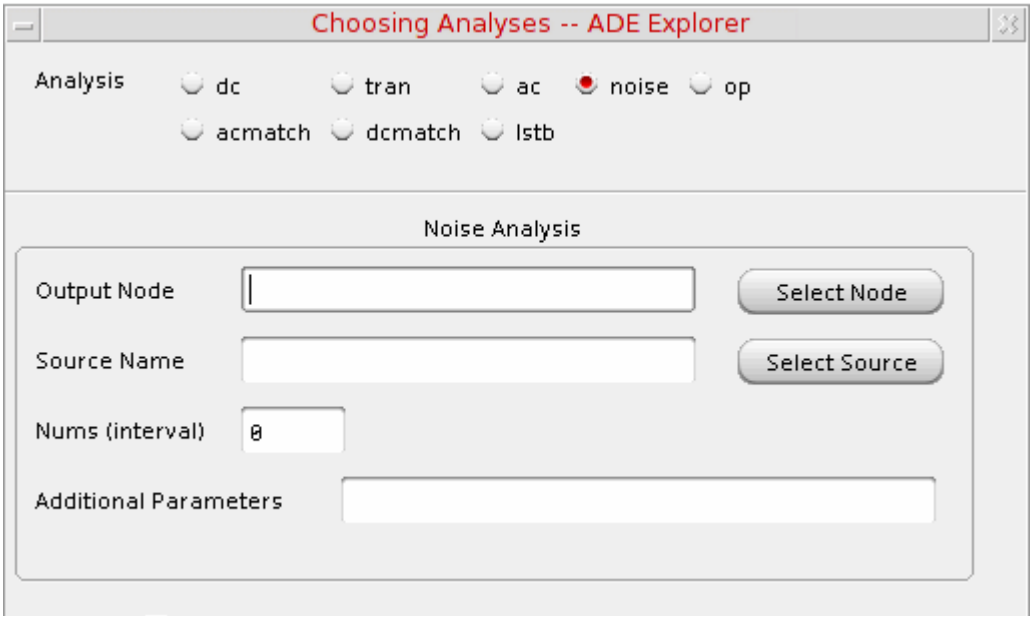
## Noise Analysis

Noise analysis linearizes the circuit about the DC operating point and computes the total-noise spectral density at the output.

Perform the following step to set up a noise analysis

➡ In the *Choosing Analyses* form, choose *noise*.

The *Choosing Analyses* form is updated to show the additional fields for noise analysis.




For more information about noise analysis, see Noise Analysis on page 288.

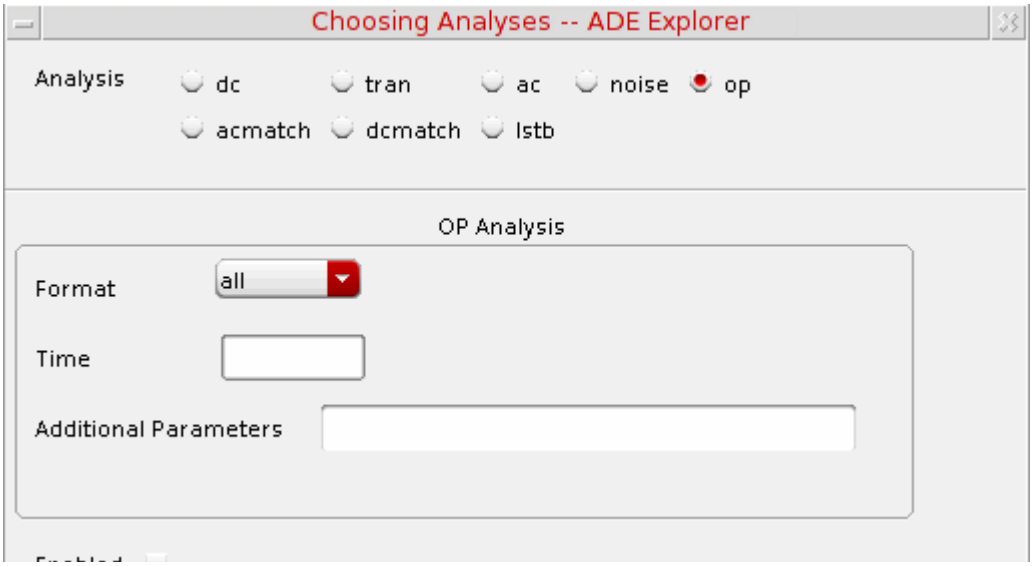

© 2023 Cadence Design Systems, Inc.

## OP Analysis

Perform the following step to set up an OP analysis:

➡ In the *Choosing Analyses* form, choose *op*.

The *Choosing Analyses* form is updated to show the additional fields for OP analysis.





© 2023 Cadence Design Systems, Inc.

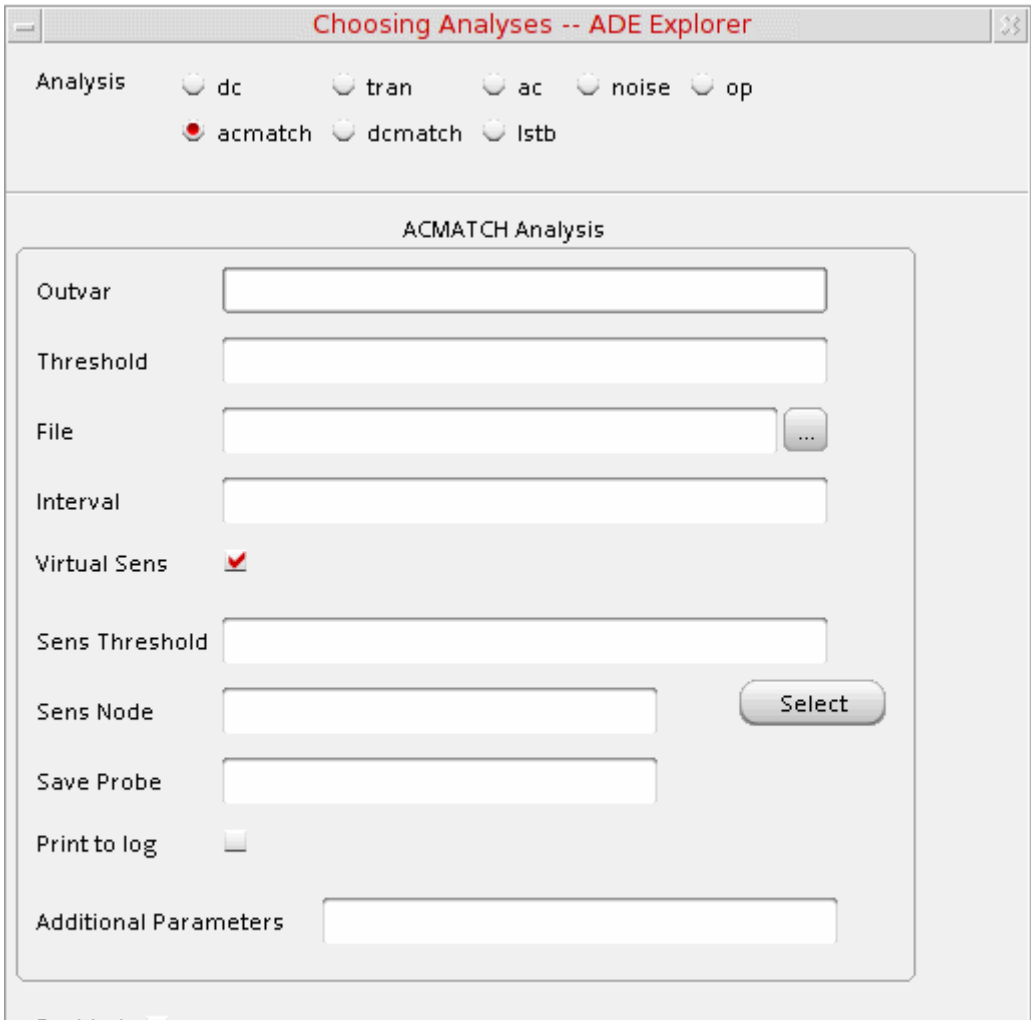
## AC Match Analysis

The *acmatch* analysis option is used to linearize the circuit around the DC operating point and computes the variations of AC responses due to statistical parameters defined in statistical block.

Perform the following step to set up an AC Match analysis:

➡ In the *Choosing Analyses* form, choose *acmatch*.

The *Choosing Analyses* form is updated to show the additional fields for AC Match analysis.




For more information about AC Match analysis, see AC Match Analysis on page 314.


© 2023 Cadence Design Systems, Inc.

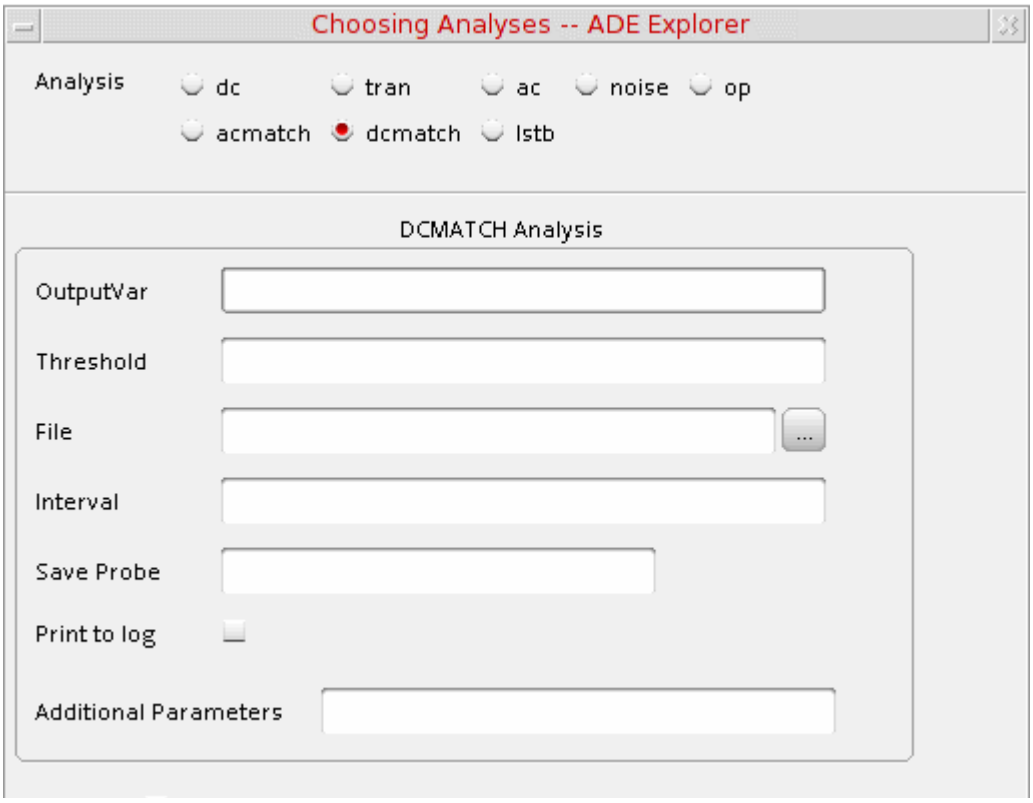
## DC Match Analysis

The *dcmatch* analysis option performs DC device mis-matching analysis for a given output. It computes the deviation in the DC operating point of the circuit caused by mismatch in the devices.

Perform the following step to set up a DC Match analysis:

➡ In the *Choosing Analyses* form, choose *dcmatch*.

The *Choosing Analyses* form is updated to show the additional fields for DC Match analysis.




For more information about DC Match analysis, see DC Match Analysis on page 309.

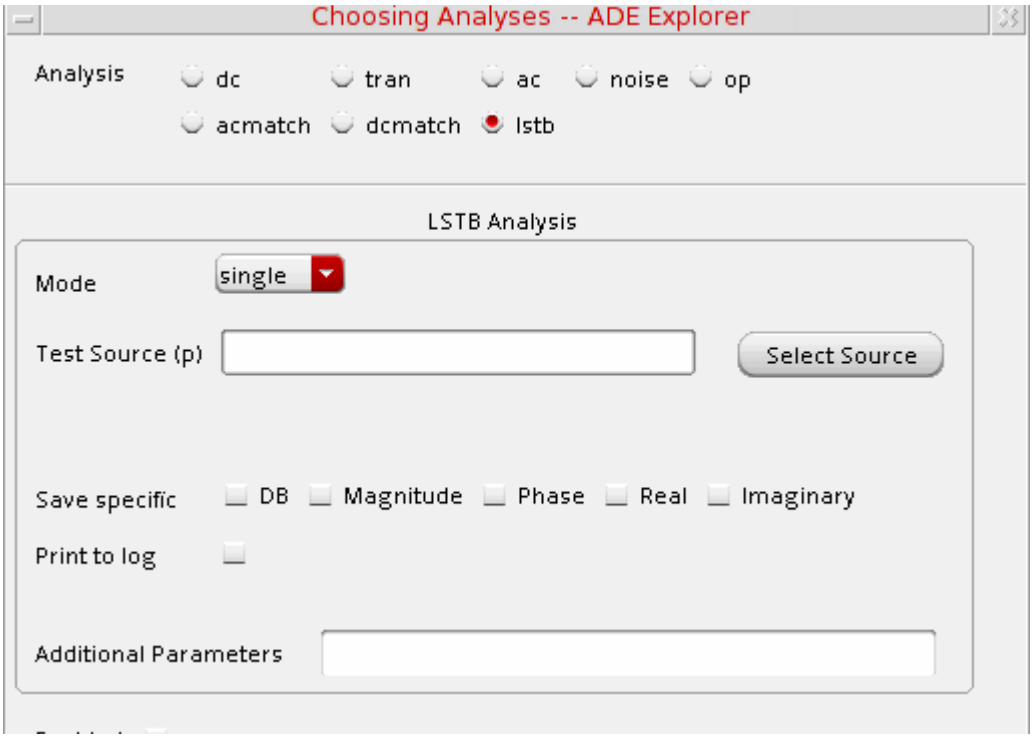

© 2023 Cadence Design Systems, Inc.

## LSTB Analysis

Perform the following step to set up LSTB Analysis:

➡ In the *Choosing Analyses* form, choose *lstb*.

The *Choosing Analyses* form is updated to show the additional fields for LSTB analysis.





© 2023 Cadence Design Systems, Inc.

# Setting Up EM/IR Analysis

The dynamic power net and signal net EM/IR capability is designed to support high-capacity and high-performance EM/IR analyses. Within the same flow, Spectre APS can be used for high accuracy EM/IR analyses, and Spectre XPS can be deployed for high capacity and high performance EM/IR analyses.

The EM/IR solution provides a set of advanced features covering power gate support, design resistor EM calculation, macro-model generation, blackboxing, what-if analysis, and static EM/IR analysis. This reduces the probability of power integrity issues in IC designs early on. It also lets you to run a quick sanity check to identify problems in a circuit schematic due to reasons like length and width of wires.

The following topics are covered in this section:

- Specifying Basic Settings
- Specifying Technology Files
- Setting Up Voltus-Fi Options
- Setting Up the Solver Method and Time Window
- Reviewing Summary Information and Adding Customized Options
- Viewing Results
- Viewing the Layout Analysis


© 2023 Cadence Design Systems, Inc.

## Specifying Basic Settings

To specify basic settings for an EM/IR analysis:

1. Choose *Setup – EM/IR Analysis*.

   The *EMIR Analysis Setup* form is displayed.

2. In the *Analysis* group box, select *Enable EMIR Analysis in Transient or DC Simulation*.
3. From the *Type* list, select one of following types of analyses.
   - *Dynamic:* calculates the IR and EM current density by performing a DC or transient simulation.
   - *Static:* evaluates IR drops and EM currents based on the specified current consumptions for subcircuit instances.


© 2023 Cadence Design Systems, Inc.

- *SPGS:* calculates pin-to-tap resistances based on the description of a DSPF file and the options set in an EMIR configuration file.

4. In the *Net/Instance* table, specify the nets or instances by selecting them from the schematic or adding them manually. Then select the analyses to be applied to them.
5. In the *DSPF* group box:
   a. Select *Enable RF Reference Flow* to enable the RF reference flow that lets you add passive devices for electromigration analysis like inductors and transformers.

      This adds a line to the EMIR configuration file and the *Summary* tab of the form. It points to the DSPF file corresponding to the nport sparam file and the hierarchical instance name of that nport in the netlist of the smartview_sparam view.

      For example, in the following line, `dspf_filename` points to the DSPF file in the smartview_sparam view and `passive_cell_inst` points to the hierarchical instance name of the nport.

      ```
      spf passive_cell_dspf=dspf_filename passive_cell_inst=[inst1 inst2]
      ```

   b. Review the list of DSPF files. Click *Open Simulation Files Setup Form* to edit this list, if required. You can also view the size of the file and status of the SPF check performed for the file.

   c. Click *Spf Checker* to use the SPF Checker utility. This checks the DSPF files, reports issues that may cause simulation problems, and creates an EMIR configuration file with recommended mapping statements. You can specify a name for this file using the field *DSPF emir .conf file* in the *DSPF File(s)* table. If this field is left empty, the DSPF file name is used as the configuration file name.

      Note the following:

      - If the `-outdir` option is used to specify the output directory name, the SPF checker will use that.
      - In case you specify a DSPF cellview in the Simulation Files Setup form, the SPF checker uses the libary/cell/view information to create separate directories for each cellview.

   d. Click *OK*.

### *Related Topics*

- EMIR Analysis Setup – *Basic Tab*


© 2023 Cadence Design Systems, Inc.

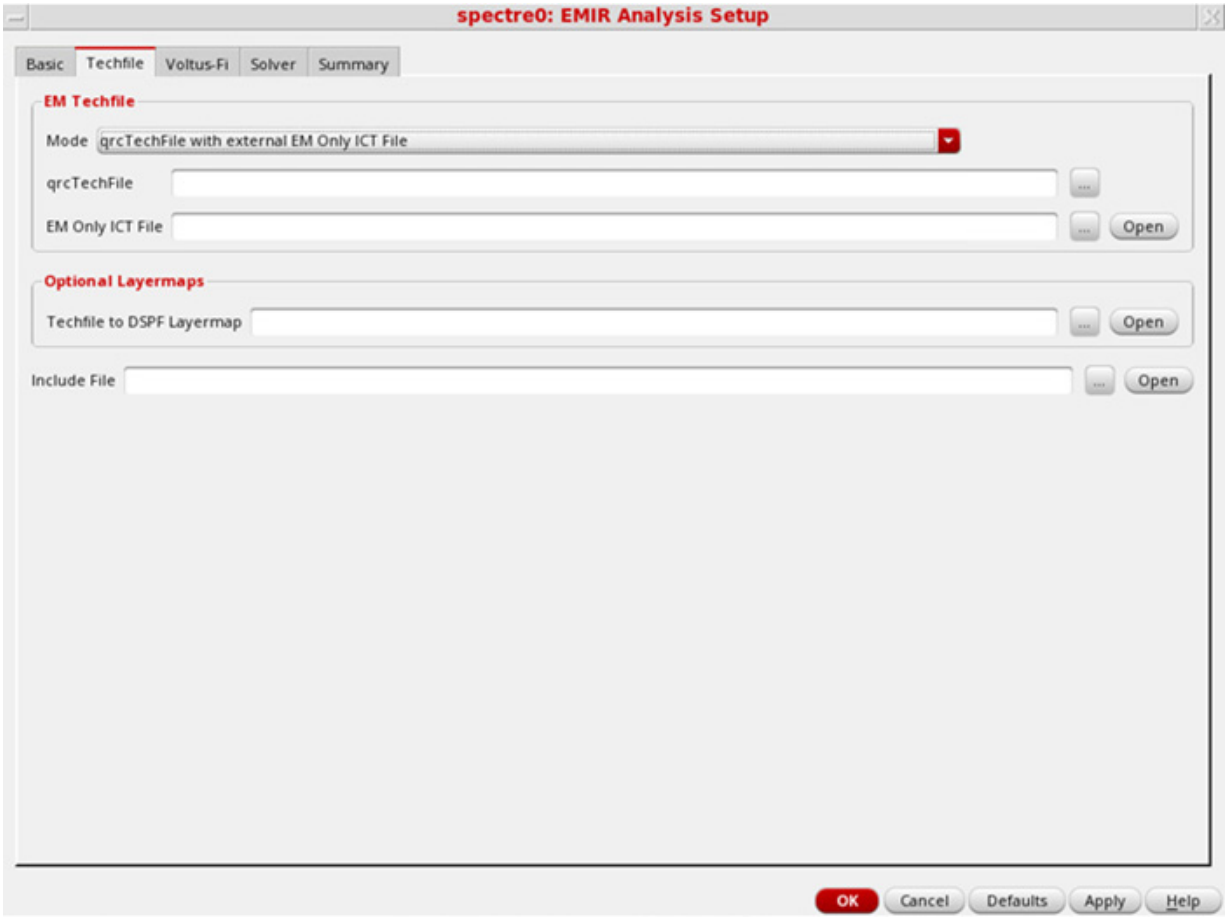
- Setting Up EM/IR Analysis for AMS Simulator(video)
- Environment Variables


© 2023 Cadence Design Systems, Inc.

## Specifying Technology Files

To specify the EM technology files,

1. Click the *Techfile* tab.




2. Select one of the following modes:
   - ❑ *qrcTechFile with external EM only ICT File*
   - ❑ *qrcTechFile with embedded emModel section*
   - ❑ *ICT File with external EM only ICT File*
   - ❑ *ICT File with embedded emModel section*
   - ❑ *emData File*


© 2023 Cadence Design Systems, Inc.

3. (Optional) In the *Techfile to DSPF Layermap* field, specify a file containing the mapping information between layer names in the xDSPF file and layer names in the technology file.
4. Specify additional technology files, if any, in the *Include File* field.

### *Related Topics*

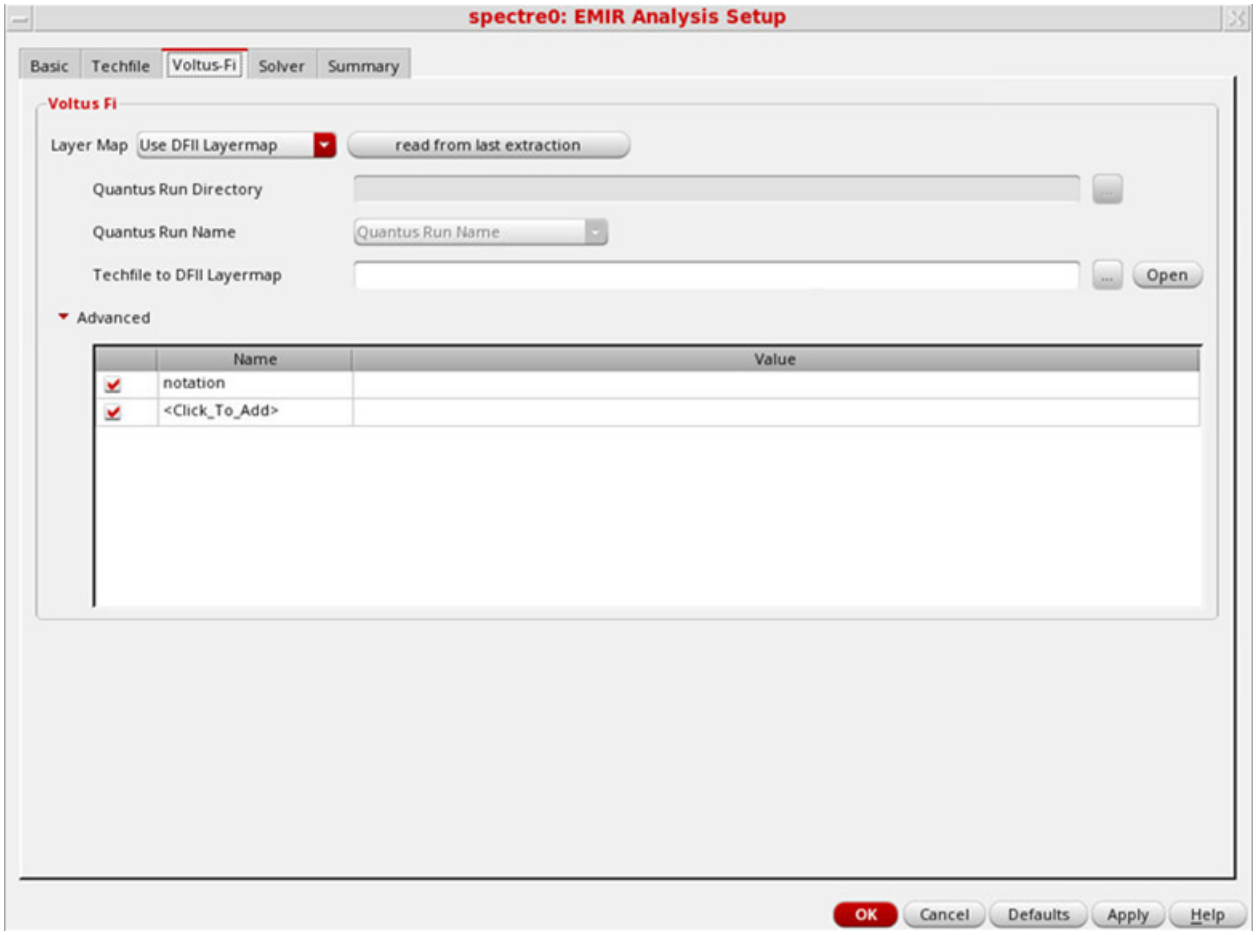
- EMIR Analysis Setup – *Techfile Tab*
- Setting Up EM/IR Analysis for AMS Simulator(video)
- Environment Variables


© 2023 Cadence Design Systems, Inc.

## Setting Up Voltus-Fi Options

To set up the Voltus-Fi options,

1. Click the *Voltus-Fi* tab.




2. In the *Voltus-Fi* tab, select one of the following layer map methods:
   - ❑ *None*
   - ❑ *Use DFII Layermap*
   - ❑ *Use Quantus Run*
3. Specify the Quantus run directory and Quantus run name
4. In the *Techfile to DFII Layermap* field, specify a file containing the mapping information between the DFII layer names and layer names in the technology file.
5. The other advanced options can be specified in the *Advanced* table.


© 2023 Cadence Design Systems, Inc.

Alternatively, you can load the Voltus-Fi setup from the previous EM/IR Quantus xDSPF run by clicking the *read from last extraction*.

### *Related Topics*

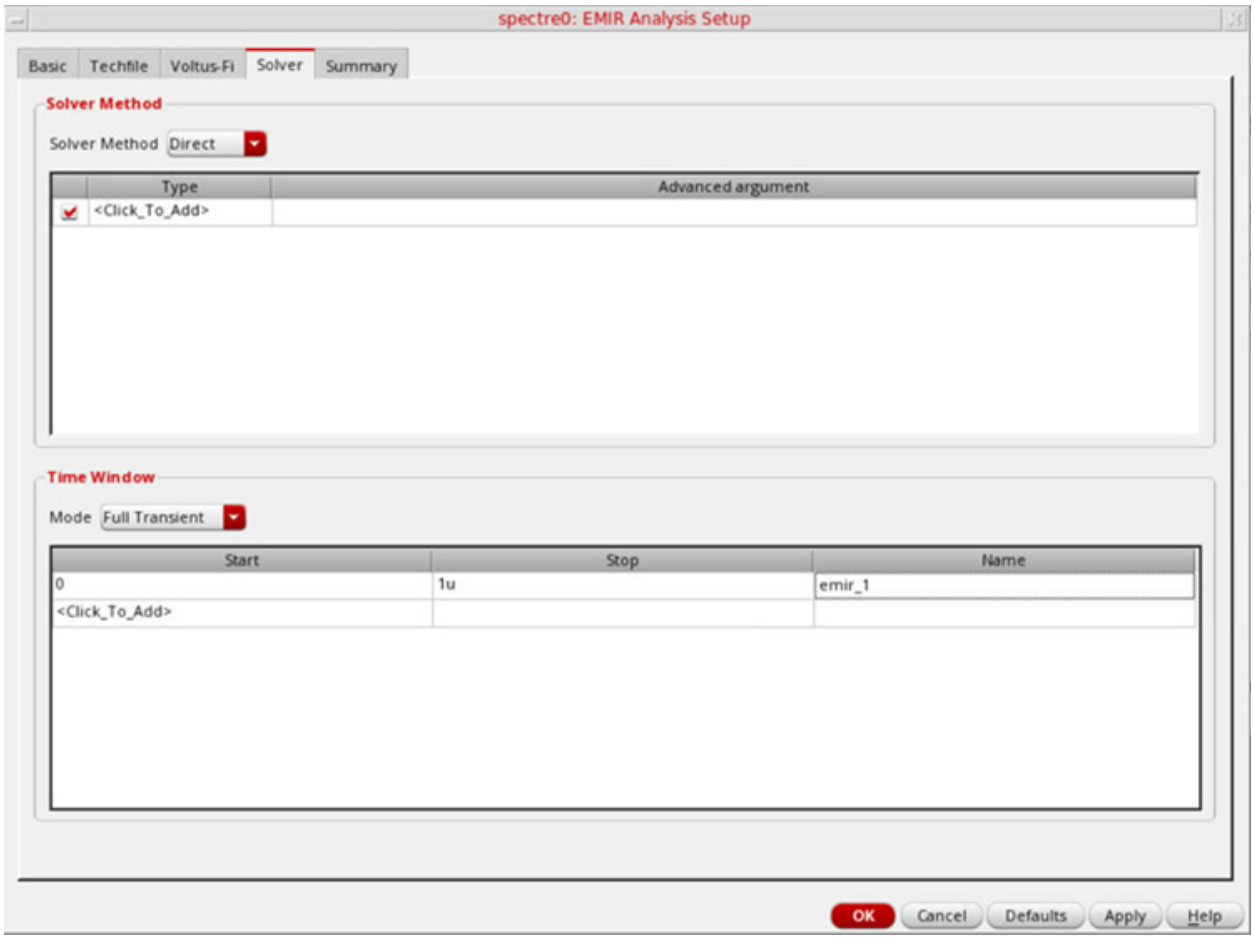
- EMIR Analysis Setup – *Voltus-Fi Tab*
- Setting Up EM/IR Analysis for AMS Simulator(video)
- Environment Variables


© 2023 Cadence Design Systems, Inc.

## Setting Up the Solver Method and Time Window

To set up the solver method and time window,

1. Click the *Solver* tab.




2. In the *Solver Method* group box, select one of the following the solver methods:

   - *Direct:* When high accuracy is needed, a brute-force simulation of the entire system—circuit plus parasitic resistances and capacitances—can be performed to accurately calculate EM/IR of any net. This approach is called the “one-step” method, where the EM/IR simulation performance and capacity are subject to the limitations of the circuit simulator being used
   - *Iterated:* To conduct EM/IR simulation on circuits with much larger power and signal nets, within a shorter time, there is an alternative approach that involves decoupling the nonlinear circuit simulation from the linear RC net analysis. You can iterate the linear RC net analysis by modifying the layout. However, the nonlinear circuit simulation is done only once. This approach is called the two-step method.


© 2023 Cadence Design Systems, Inc.

3. Any advanced arguments related to the solver can be defined in the table below the *Solver Method* drop-down list.

   Note the following:

   - You can view the solver method setting in the Summary Information table, only when the solver method is updated.
   - On selecting *Iterated* as the solver method, the argument, `rcr=selected` gets listed in this table.
   - The RCR report, `*.rpt_rcr` is generated only if the specified solver method is *Iterated*.

4. In the *Time Window* group box, select one of the following modes:

   - *Full Transient*: The EM/IR analysis is run over the entire simulation period.
   - *Time Window*: The EM/IR analysis is run for a specific time interval.

     In the *Time Window* table, you can specify the time period in the *Start* and *Stop* columns and the name of the time window in the *Name* column.

***Related Topics***

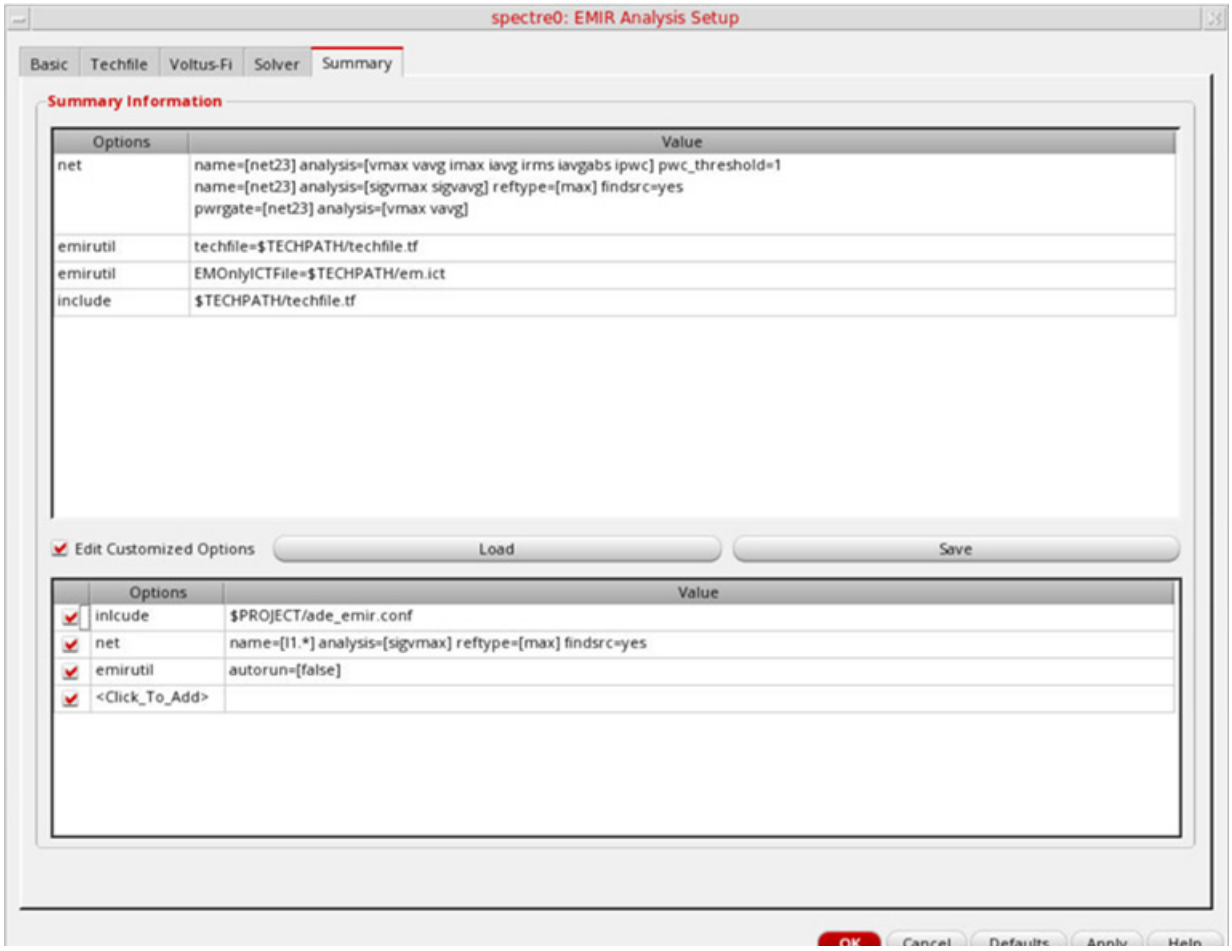
- EMIR Analysis Setup – *Solver Tab*
- Setting Up EM/IR Analysis for AMS Simulator(video)
- Environment Variables


© 2023 Cadence Design Systems, Inc.

## Reviewing Summary Information and Adding Customized Options

To review the summary information table and add customized options,

1. Click the *Summary* tab.




2. In the upper-half of the tab, review all the settings specified in the rest of the tabs.
3. Click *Load* to import the EM/IR analysis settings from an existing configuration file to the customized options table in the lower-half of the tab.
4. Select *Edit Customized Options* to add, edit, or delete options supported by Spectre for EMIR analysis. You can also add these options through the SKILL function `asiSetEMIROptionVal`.
5. Click *Save* to save the settings from the customized options table to a configuration file.
6. Click *OK*.


© 2023 Cadence Design Systems, Inc.

### *Related Topics*

- EMIR Analysis Setup – *Summary Tab*
- Setting Up EM/IR Analysis for AMS Simulator(video)
- Environment Variables
- asiSetEMIROptionVal

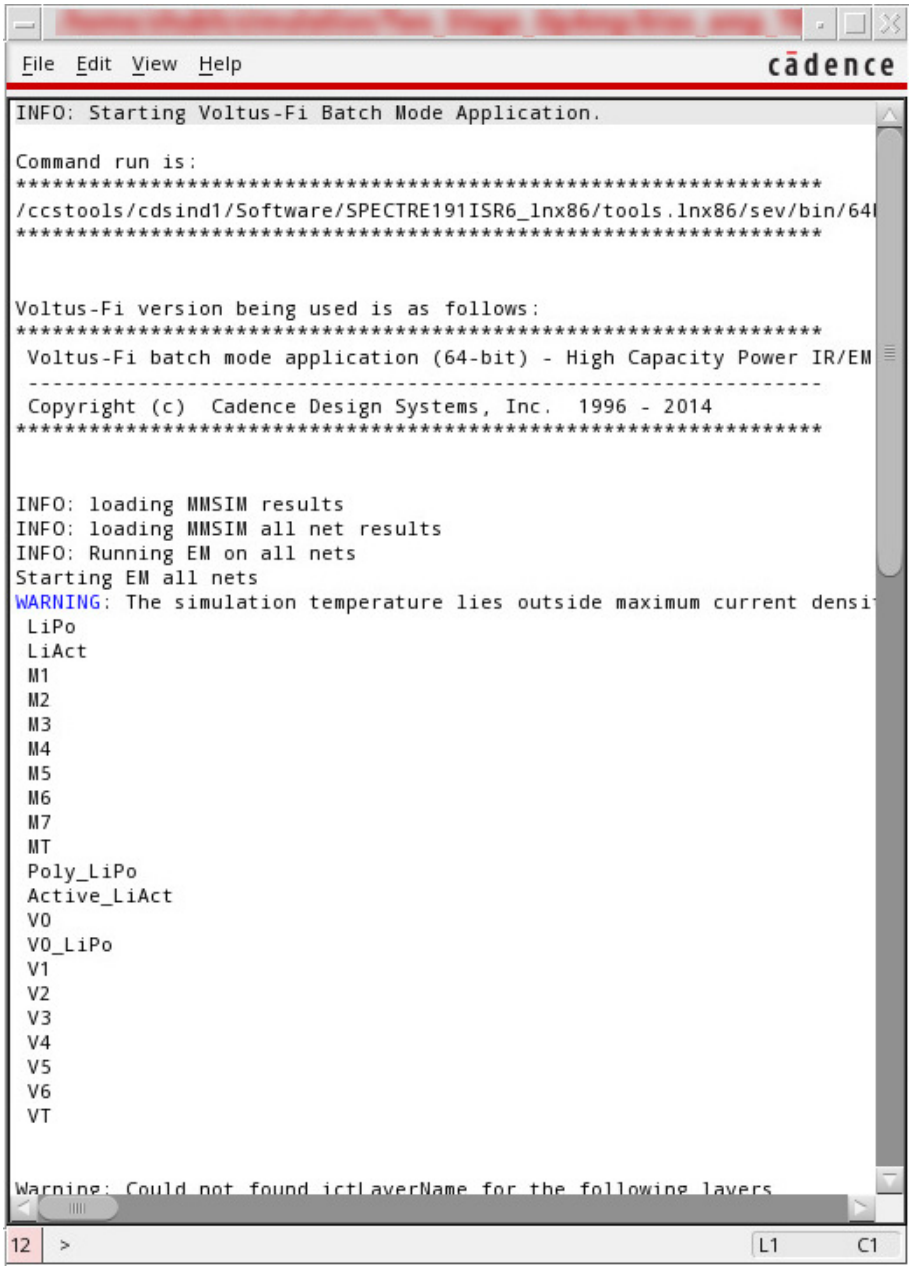

© 2023 Cadence Design Systems, Inc.

## Viewing Results

After running a simulation, in the ADE maestro view, choose *Results — EM/IR Data* to view the output results.

The EM/IR data includes:

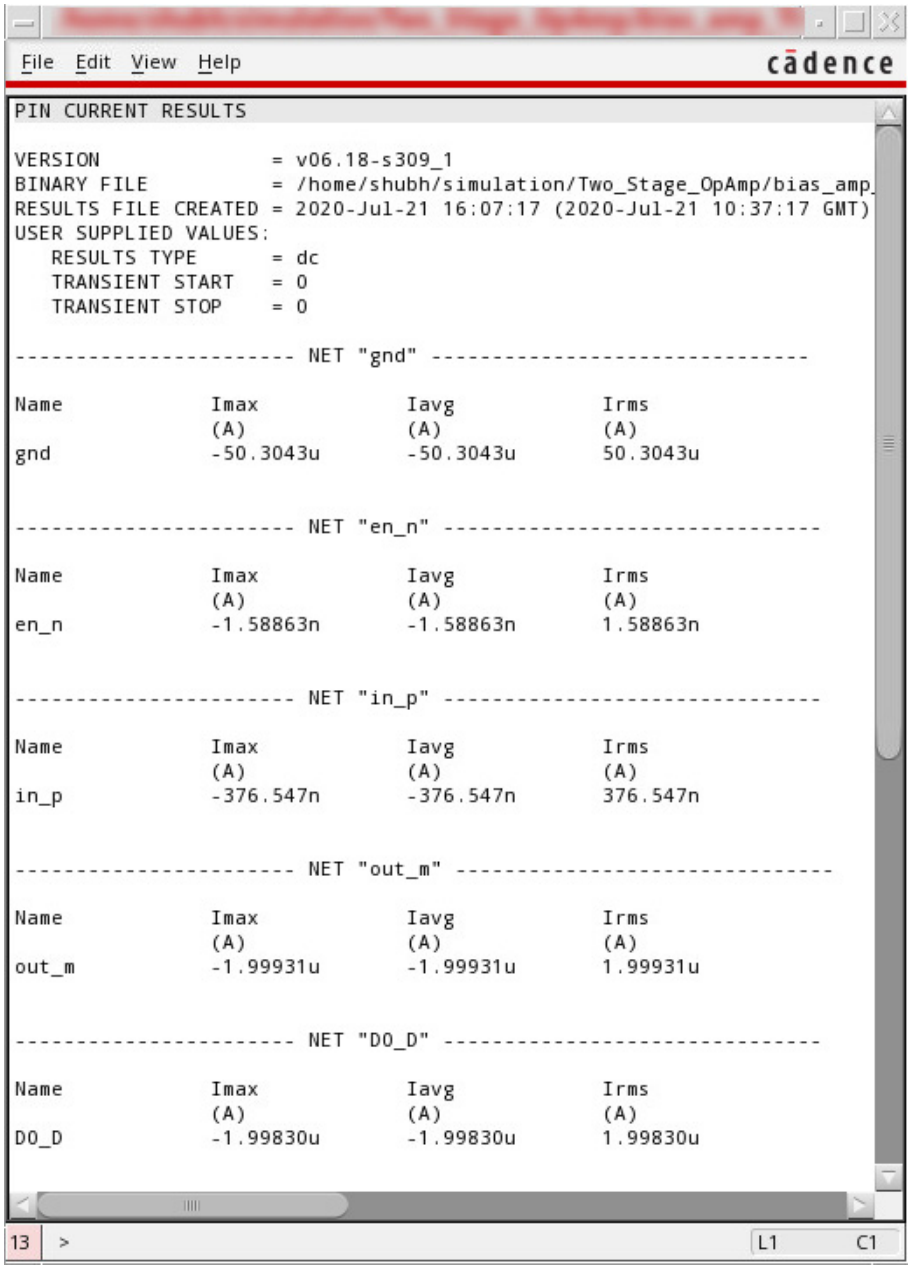
- EM/IR Log: the EM/IR log file that contains the messages issued by the tool and the commands used during the simulation run.





© 2023 Cadence Design Systems, Inc.

- Report: the EM/IR report created after the simulation run.




This report displays the following results:

- Pin current results
- Voltage drop results
- Electomigration analysis results


© 2023 Cadence Design Systems, Inc.

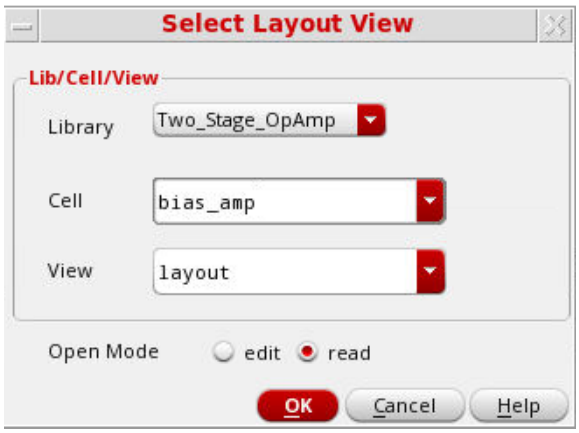
- Layout Analysis: displays the *IR/EM Results* window for the specified layout view. This contains the settings and file paths as specified in the EM/IR Analysis.


© 2023 Cadence Design Systems, Inc.

## Viewing the Layout Analysis

1. To view the layout analysis, choose *Results — EM/IR Data — Layout Analysis.*

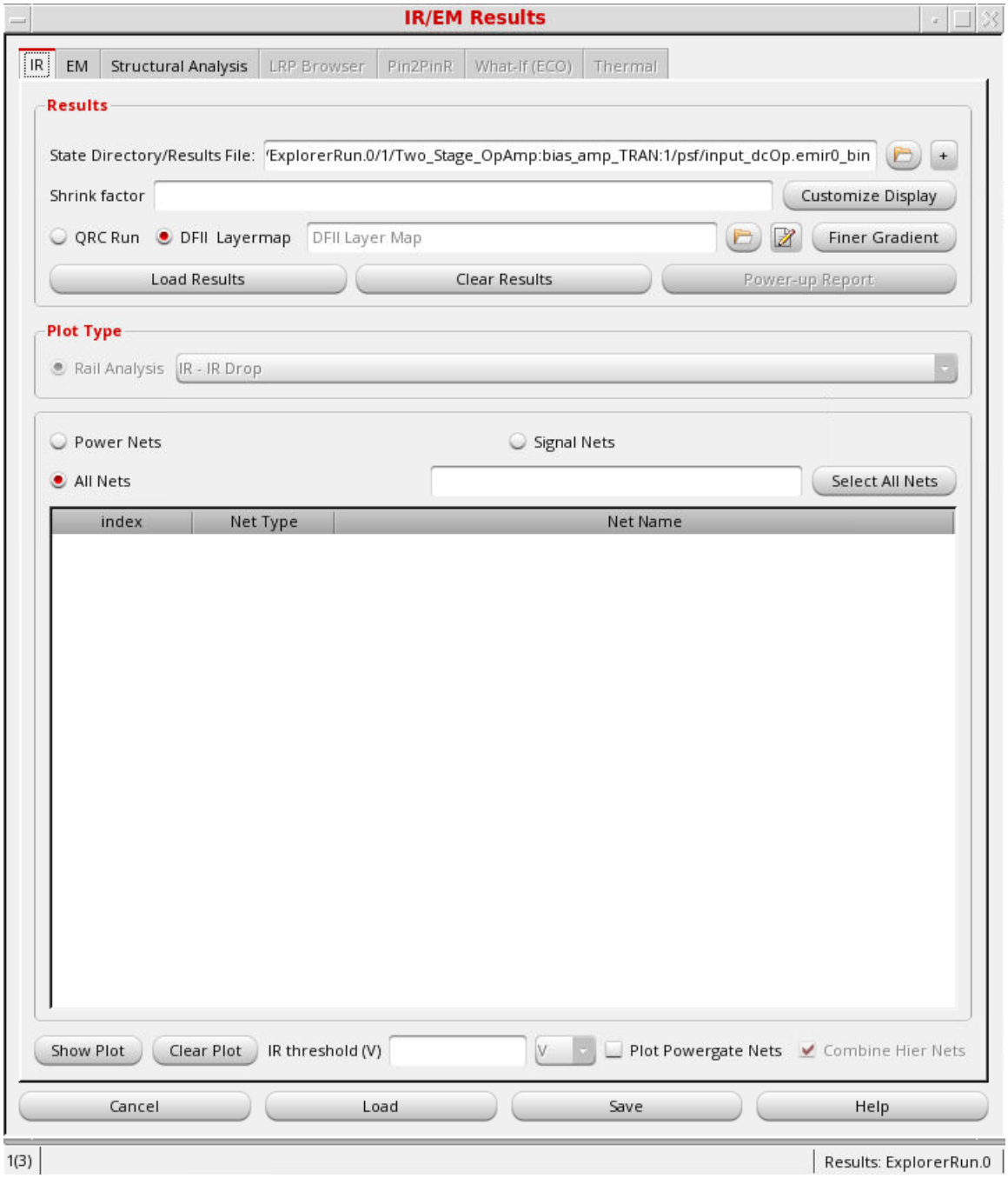
   The *Select Layout View* form is displayed.




2. Specify the corresponding layout view and click *OK*.


© 2023 Cadence Design Systems, Inc.

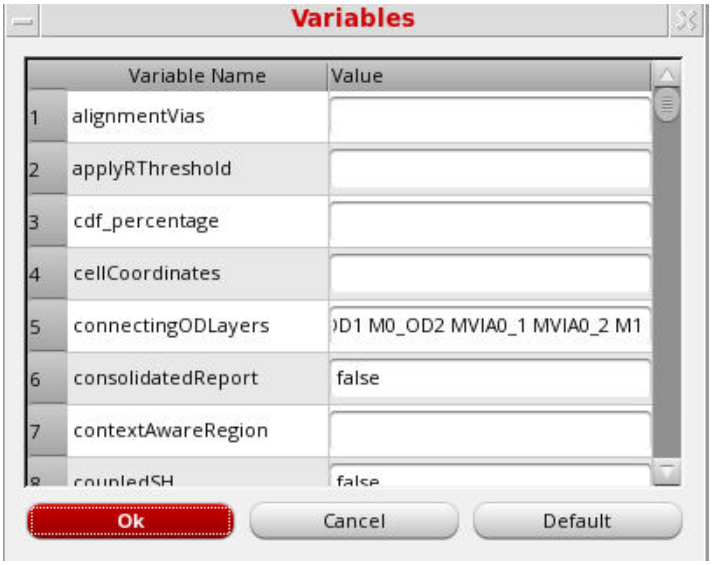
The *IR/EM Results* window is displayed where the settings and file paths are already specified according to the settings in the EM/IR Analysis Setup form.





© 2023 Cadence Design Systems, Inc.

3. To view the parameters specified in the *EM/IR Analysis Setup* form, click the *Variables* option in the *EM* tab of the I*R/EM Results* window.





© 2023 Cadence Design Systems, Inc.

# EMIR Analysis Setup Form Description

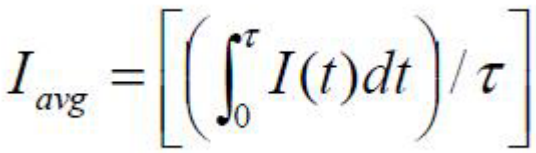
## EMIR Analysis Setup – Basic Tab

| Field | Description |
|---|---|
| **Analysis** | |
| *Enable EMIR Analysis in Transient or DC Simulation* | Enables EM/IR analysis during the simulation run. |
| *Dynamic* | Estimates the IR and EM current density by performing a DC or transient simulation. |
| *Static* | Enables you to evaluate IR drops and EM currents based on the specified current consumptions for subcircuit instances without running a transient or DC simulation. The specified currents are distributed to the tap devices based on the width and length ratios of devices in the design. The IR drop and EM current analysis is performed based on the current at each tap device. |
| *SPGS* | Used to calculate pin-to-tap resistances based on the description of a DSPF file and the options set in an EM/IR configuration file. The resistances calculated by SPGS are electrically-equivalent resistances, and not the summation of resistors.<br>The calculation assumes that all pins are connected together to form a global pin. After calculation, the resistance between the global pin and all taps is generated and listed based on the significance of their values. |
| *Net Name* | Name of the net that is selected from the schematic or added manually. |
| **Advanced IR Drop Analysis** | |
| *filter moscap* | Filters out the tap nodes connected to MOSFET devices. |
| *short_res_layer* | Specifies the DSPF layer that needs to be shorted. |
| *short* | Specifies that the resistors with `R<rshort` need to be shorted during static power grid analysis. |


© 2023 Cadence Design Systems, Inc.

| Field | Description |
|---|---|
| *LayerName* | Name of the layers to be analyzed. |
| *Pin/Tab/Sub Conversion Options* | Displays the *SPGS Options* form. |
| **SPGS Options** | |
| *tap2sub* | Converts taps to subnodes. |
| *tap2pin* | Converts taps to pins. |
| *sub2tap* | Converts subnodes to taps. |
| *sub2pin* | Converts subnodes to pins. |
| *pin2sub* | Converts pins to subnodes. |
| *pin2tap* | Converts pins to taps. |
| *Net Name* | Name of the net selected from the schematic. |
| *Include* | Name of the tap, pin, or subnode to be included for conversion process. |
| *Exclude* | Name of the tap, pin, or subnode to be excluded from the conversion process. |
| *Layer* | Layer containing the specified tap, pin, or subnode. |
| **Analysis Table** | |
| *Net/Instance* | Nets or instances for which the analysis is to be performed. Either select them from the schematic or add them manually. |
| *IR Max* | Maximum IR drop. |
| *IR Avg* | Average IR drop. |
| *EM Max* | Maximum voltage drop value. |
| *EM Avg* | Average voltage drop value. |
| *EM Rms* | Root mean square EM analysis. |

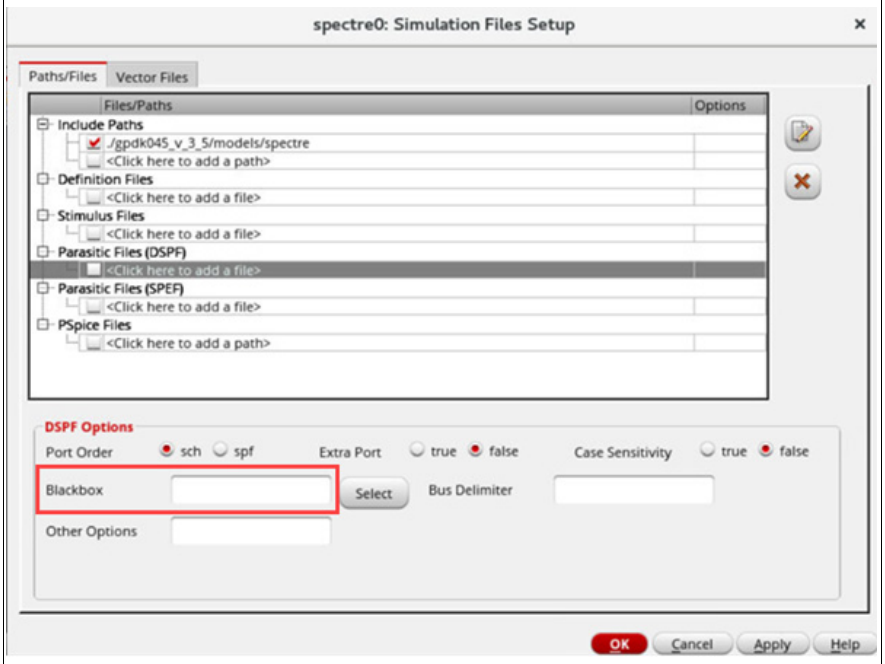

© 2023 Cadence Design Systems, Inc.

| Field | Description |
|---|---|
| *EM Avgabs* | Calculates the Current Density violations based on the average of the absolute current for metal lines, vias, and contacts.<br>  <br>where `T` is time and `I(t)` is value of current. |
| *EM AC Peak/pwc* | Breaks down the waveform into multiple pulses and calculates their AC Peak value and EM current for violation visualization.<br>The `pwc_threshold` defines the current threshold for finding the start point of the pulse. Any value below pwc_threshold is considered as zero. |
| *sig vmax* | Maximum voltage drop for the signal net. |
| *sig vavg* | Average voltage drop for the signal net. |
| *reftype* | ■ *avg* - reference voltage for IR drop is the average voltage of all subnodes at every time point. This is the default selection.<br>■ *max* - reference voltage for IR drop is the maximum voltage of all subnodes at every time point.<br>■ *min* - reference voltage for IR drop is the minimum voltage of all subnodes at every time point.<br>■ *pin* - There are two possible cases:<br>❑ When the `findsrc` option is set to `yes`, the reference voltage for IR drop is taken from vsource, which is traced back from the `*P\|` node.<br>❑ When the `findsrc` option is set to `no`, the reference voltage for IR drop is taken from the `*P\|` node. |
| *find src* | Uses voltage from subnode that is connected to vsource and automatically traces through the design or source resistors. |


© 2023 Cadence Design Systems, Inc.

| Field | Description |
|---|---|
| *PwrGate vmax* | Enables power gate handling. You need to specify the power supply net driving the power gate and the internal power supply net driven by the powergate. If one power supply net drives multiple power gates, then you need to specify one statement with all internal power supply nets. |
| *PwrGate vavg* | Allows to specify the power supply net driving the power gate and the internal power supply net driven by the powergate. If one power supply net drives multiple power gates, then specify one statement with all internal power supply nets. |
| *Power Gate* | Enables power gate handling.You need to specify the power supply net driving the power gate (vsource connected) and the internal power supply net driven by the power gate. If one power supply net drives multiple power gates, then you need to specify one statement with all internal power supply nets. |
| *Additional Arguments* | Any additional arguments can be added here. |
| *Static Current File* | Specifies the path to a static current file. This file contains the user-specified current estimates. |
| **DSPF** | |
| **DSPF Files(s)** | |
| *Open Simulation Files Setup Form* | Displays the *Simulation Files Setup* form, which you can use to edit the list of DSPF files to be used. |


© 2023 Cadence Design Systems, Inc.

| Field | Description |
|---|---|
| *Spf Checker* | Analyzes the DSPF files, reports problems that may cause simulation problems, and creates an EMIR configuration file with recommended mapping statements.<br><br>After the SPF checker is run, the name of the blackboxes detected in the DSPF files are automatically added to the *Blackbox* field in the *DSPF Options* section of the *Simulation Files Setup* form.<br><br>  |
| *DSPF File* | DSPF files contain the post-layout data of the design, including fingered devices and net parasitics, with the parasitic and instance sections describing the circuit to be analyzed. |
| *Options* | Lets you add spf checker options. |
| *Size* | Size of the DSPF file. |
| *Status* | Status of the spf check. |


© 2023 Cadence Design Systems, Inc.

## EMIR Analysis Setup – Techfile Tab

| Field | Description |
|---|---|
| **EM Techfile** | |
| *qrcTechFile with external EM only ICT File* | Specifies a QRC technology file and an EM only ICT file. |
| *qrcTechFile with embedded emModel section* | Specifies a QRC technology file. |
| *ICT File with external EM only ICT File* | Specifies an ICT file and an EM only ICT file. |
| *ICT File with embedded emModel section* | Specifies an ICT file. |
| *emData File* | Specifies an EM data file. |
| *qrcTechFile* | Contains the current limits for each process layer. This file is usually available through a foundry. |
| *EM Only ICT File* | Provides the process and the EM model information for EM analysis. This is a text-based file, which means you can use any text editor to enter information in this file |
| *ICT File* | Specifies an editable text-based file with EM reliability rules for EM analysis. |
| *EM Data File* | Specifies the technology information, such as current density limits, and provides a mapping between the layers for highlighting. |
| **Optional Layermaps** | |
| *Techfile to DSPF Layermap* | File containing mapping information between layer names in the xDSPF file and layer names in the technology file. |
| *Include File* | Allows to add any additional information.<br>For example, you can set up canonical poly resistance EM checking. |


© 2023 Cadence Design Systems, Inc.

## EMIR Analysis Setup – Voltus-Fi Tab

| Field | Description |
|---|---|
| **Voltus Fi** | |
| *None* | No layer map method to be used. |
| *Use DFII Layermap* | This layer map is the APS/XPS-to-DFII layer map. When this mode is specified, the plots displayed in the layout show solid shape highlighting.<br>This is the default layer map method. |
| *Use Quantus Run* | Plots are displayed based on the shape database generated by Quantus. |
| *read from last extraction* | Loads the voltus-fi setup from the previous EM/IR Quantus xDSPF run. |
| *Quantus Run Directory* | Specifies the Quantus run directory. |
| *Quantus Run Name* | Specifies the Quantus run name. |
| *Techfile to DFII Layermap* | File containing mapping information between DFII layer names and layer names in the technology file. |
| **Analysis table** | |
| *autorun* | Runs `emirutil` automatically to generate a text report. |
| *report* | Reports the current of the resistor in ampere (A). |
| *notation* | Notation for the text and html reports. |
| *geounit_xy* | Defines the scaling of DSPF geometry parameters X and Y. |
| *geounit_wl* | Defines the scaling of DSPF geometry parameters W and L. |
| *idirn* | Prints the current direction in the EM report (Current Direction column). For metal resistors, the current direction is printed. That is, W means that the current is flowing from east to west. For via resistors, the direction from and to layer is printed. |
| *ir_in_em* | Reports the maximum IR drop value for each metal resistor or via in the EM current report. It adds a “Max IR” column in the EM report. |
| *Tlife* | Specifies the lifetime for which the EM analysis is to be performed. |


© 2023 Cadence Design Systems, Inc.

| Field | Description |
|---|---|
| *Tsim* | Specifies the value of temperature in the netlist. |
| *Tj* | Specifies the temperature to be used for EM analysis. |
| *dynamicACPeak* | Specifies that the current density violations should be calculated based on the peak AC current for metal lines, vias, and contacts. It is applied to periodic AC or pulsed DC signals. |
| *view* | Specifies the layout. |
| *cds.lib* | Specifies the path of the cds.lib. It is used to specify the path when the cds.lib is not in the current working directory. |
| *includeFile* | Allows to include any additional files. |

## EMIR Analysis Setup – Solver Tab

| Field | Description |
|---|---|
| **Solver Method** | |
| *Direct* | When high accuracy is needed, a brute-force simulation of the entire system—circuit plus parasitic resistances and capacitances—can be performed to accurately calculate EMIR of any net. This approach is called the one-step method, where the EMIR simulation performance and capacity are subject to the limitations of the circuit simulator being used. |
| *Iterated* | To conduct EM/IR simulation on circuits with much larger power and signal nets, within a shorter time, there is an alternative approach that involves decoupling the nonlinear circuit simulation from the linear RC net analysis.<br>You can iterate the linear RC net analysis by modifying the layout. However, the nonlinear circuit simulation is done only once. This approach is called the two-step method. The decoupling of the linear RC nets from the nonlinear circuit is not mathematically equivalent to the original design and inaccuracies might be introduced but it provides the benefit of simulation performance and capacity. |


© 2023 Cadence Design Systems, Inc.

| Field | Description |
|---|---|
| **Time Window** | |
| *Full Transient* | EMIR analysis is run over the entire simulation period. |
| *Time Window* | You can provide the Start and Stop times |

### EMIR Analysis Setup – Summary Tab

| Field | Description |
|---|---|
| **Summary Information** | |
| *Edit Customized Options* | Lets you edit the EM/IR options listed in the table in the lower-half of the *Summary* tab. |
| *Load* | Loads the EM/IR analysis settings from an existing configuration file to the customized options table. |
| *Save* | Saves the settings from the customized options table to a configuration file. |

### Setting Up EM/IR Analysis for AMS Simulator

Video

For details on setting up AMS EM/IR analysis, view the following videos on Cadence Online Support:

- Introducing AMS EM/IR Analysis
- Setting up and running AMS EM/IR Analysis in ADE

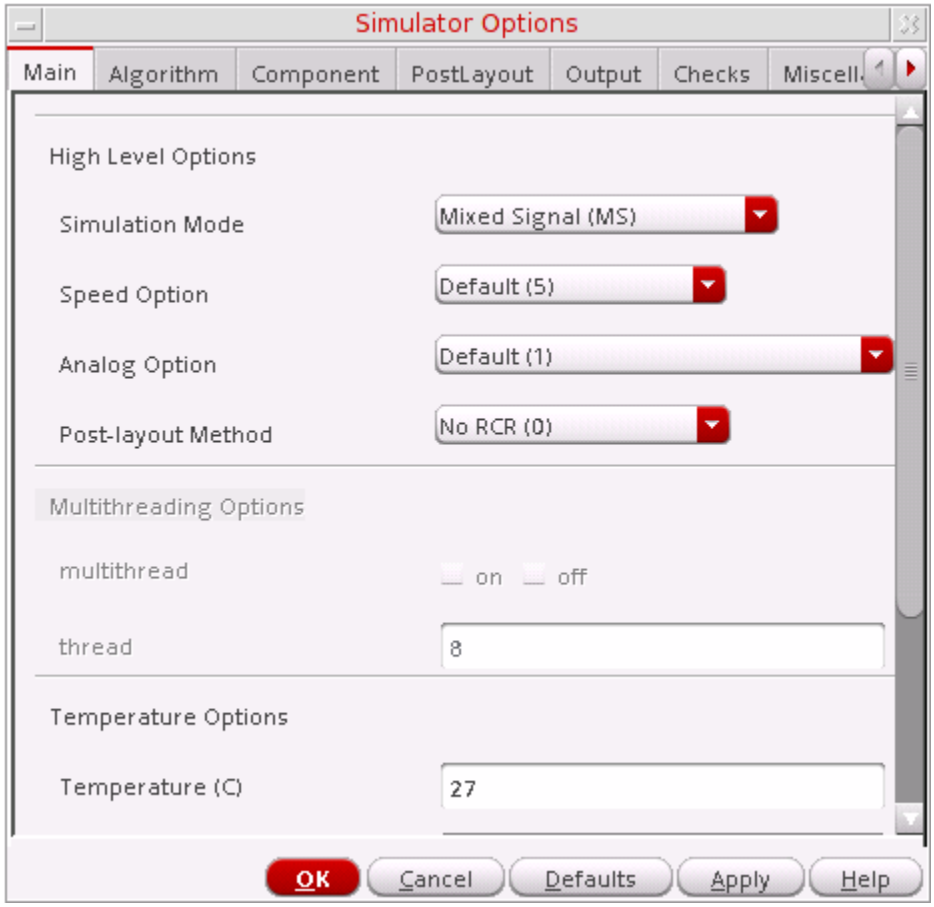
# Running Advanced Analysis Simulations

Perform the following steps to run a Virtuoso UltraSim simulator advanced analysis:

1. Choose *Simulation – Options – Analog*.


© 2023 Cadence Design Systems, Inc.

The *Simulator Options* form appears.




**Note:** You can set output format using the *Output* tab.

There are eight advanced analysis options in the Advanced Checks. The advanced checks can be set using *Checks* tab in the *Simulator Options* form:

- Timing Analysis on page 384
- Power Analysis on page 393
- Power Checking Analysis on page 394
- Design Checking Analysis on page 396
- Active Node Checking Analysis on page 398
- Node Activity Analysis on page 399
- Node Connectivity Analysis on page 399


© 2023 Cadence Design Systems, Inc.

- Reliability Analysis on page 400
- Power Network Solver on page 401

2. Choose the appropriate advanced analysis and set the options in the corresponding analysis forms.

### Timing Analysis

In Timing analysis, you can perform the following checks on signals:

- Setup Check on page 386
- Hold Check on page 388
- Pulse Width Check on page 390
- Timing Edge Check on page 392

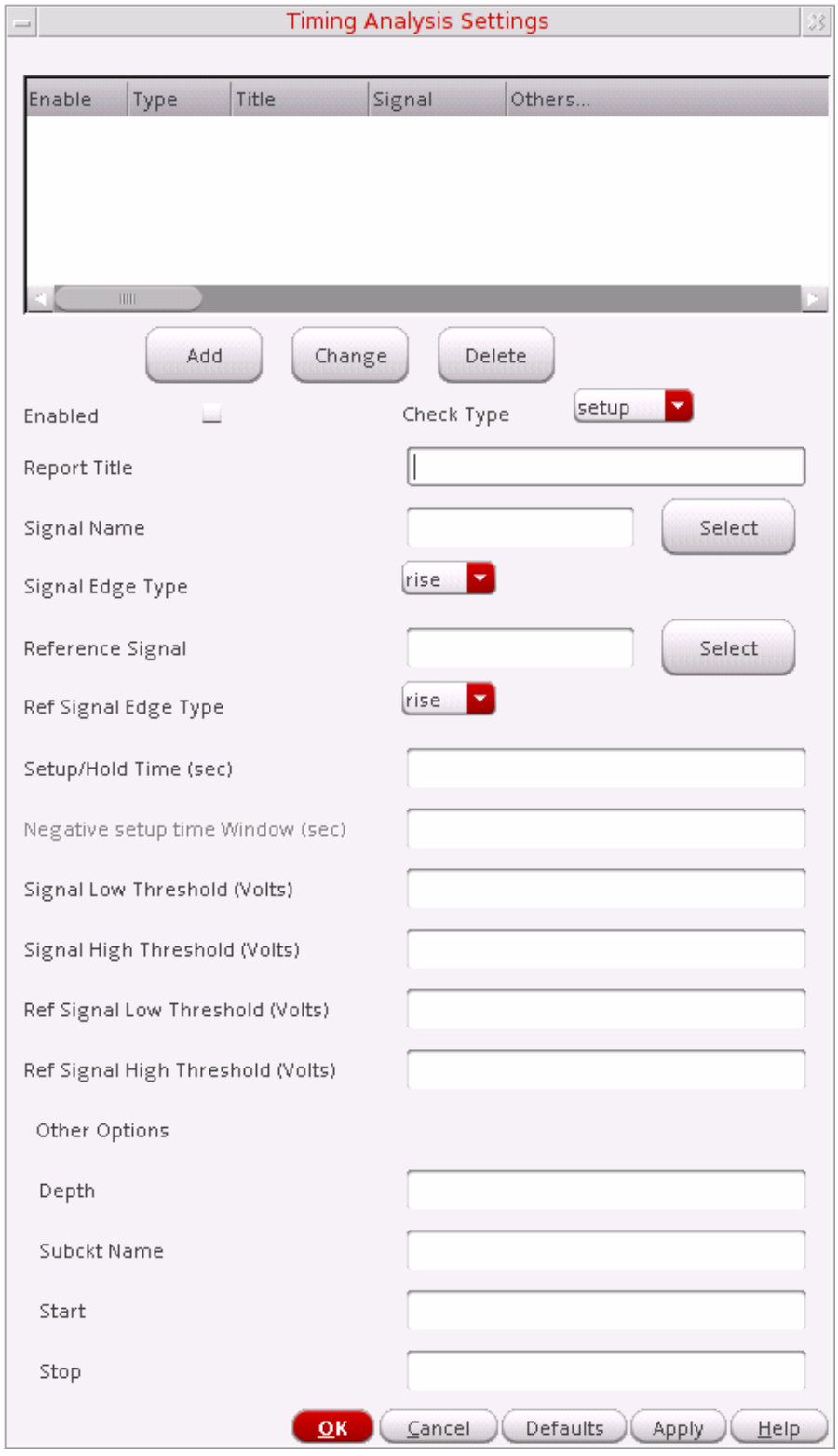

© 2023 Cadence Design Systems, Inc.

To view a timing analysis, choose *Results – Print – Advanced Analysis Results – Timing Analysis* in ADE Explorer.

For more information about the timing analysis settings, refer to Chapter 7, Virtuoso UltraSim Advanced Analysis in the *Virtuoso UltraSim Simulator User Guide*.


© 2023 Cadence Design Systems, Inc.

### *Setup Check*



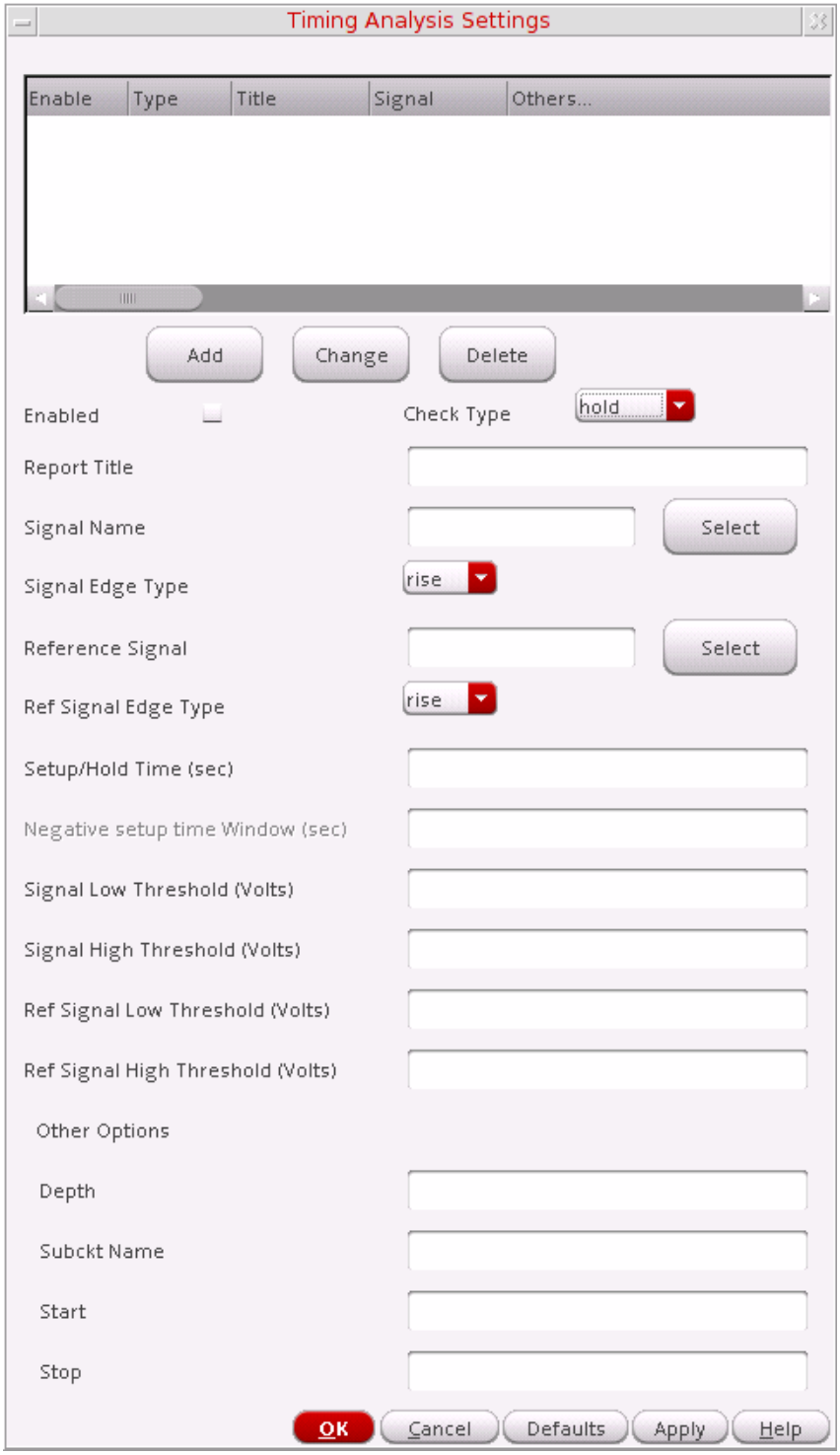


© 2023 Cadence Design Systems, Inc.

1. Choose *setup* in the *Check Type* drop-down list and adjust the timing analysis settings as needed.
2. Click *Add* to add a setup check.


© 2023 Cadence Design Systems, Inc.

### *Hold Check*





© 2023 Cadence Design Systems, Inc.

1. Choose *hold* in the *Check Type* drop-down list and adjust the timing analysis settings as needed.
2. Click *Add* to add a hold check.


© 2023 Cadence Design Systems, Inc.

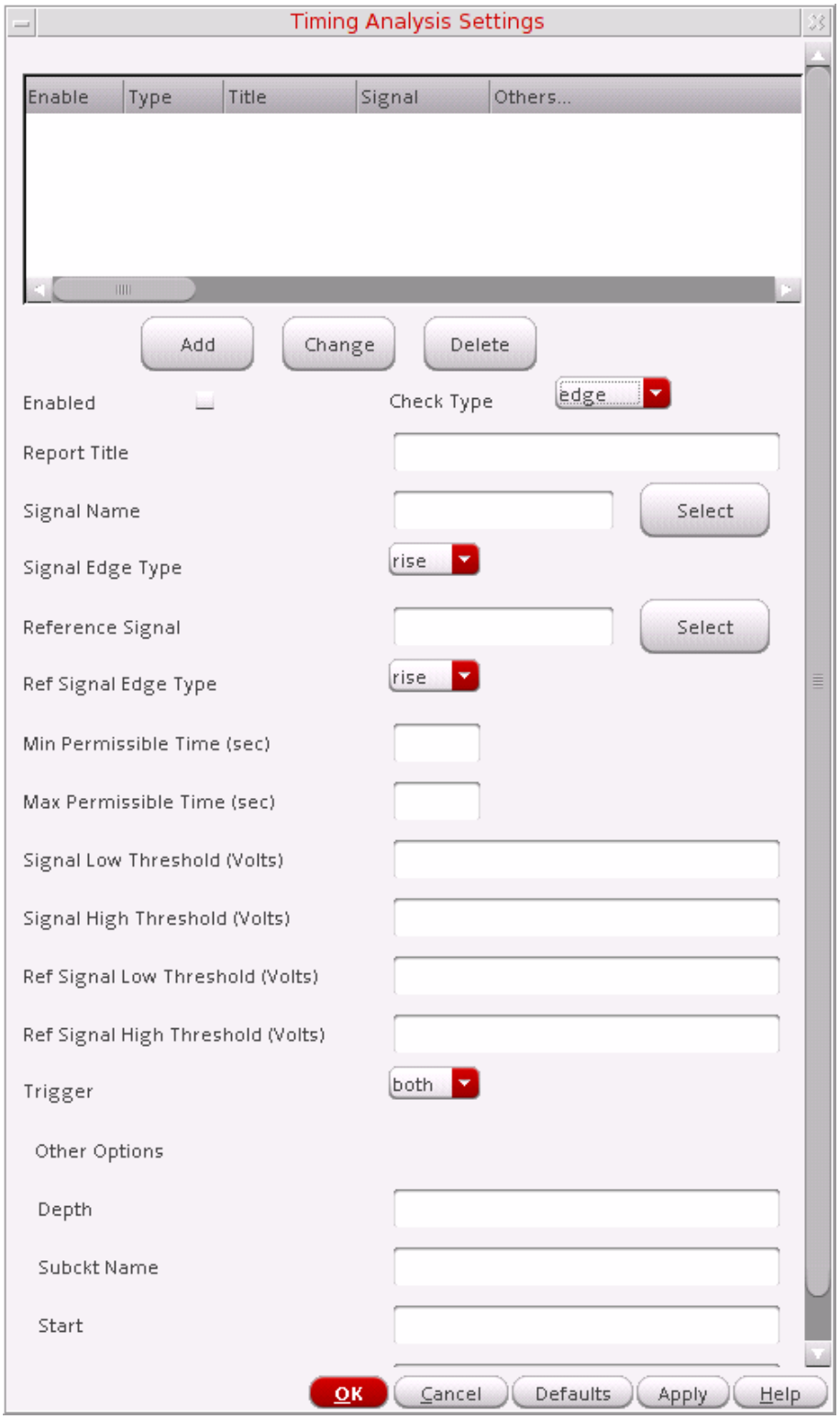
### *Pulse Width Check*

© 2023 Cadence Design Systems, Inc.

1. Choose *pulsew* in the *Check Type* drop-down list and adjust the timing analysis settings as needed.
2. Click *Add* to add a pulse width check.


© 2023 Cadence Design Systems, Inc.

### *Timing Edge Check*



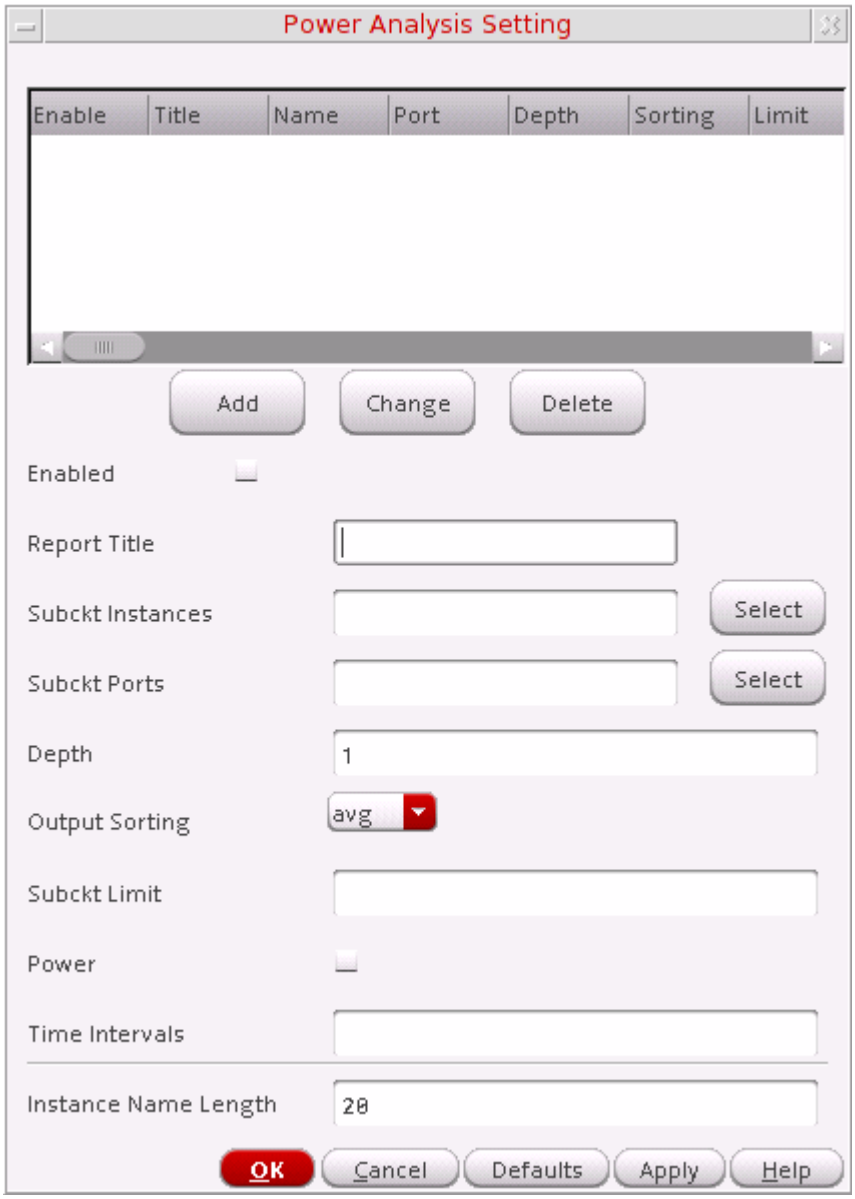


© 2023 Cadence Design Systems, Inc.

1. Choose *edge* in the *Check Type* drop-down list and adjust the timing analysis settings as needed.
2. Click *Add* to add an timing edge check.

### Power Analysis

Power analysis reports the power consumed by each element and subcircuit in the circuit.




1. Adjust the power analysis settings as needed.


© 2023 Cadence Design Systems, Inc.

2. Click *Add* to add a power analysis check.

Power analysis results can be viewed from the ADE Explorer window using *Results – Print – Advanced Analysis Results – Power Analysis*.

For more information about the power analysis settings, refer to *Virtuoso UltraSim Advanced Analysis* chapter in the *Virtuoso UltraSim Simulator User Guide*.

### Power Checking Analysis

Based on the specified element list (current threshold, over current duration time, and checking windows), the Virtuoso UltraSim simulator reports in a `.pcheck` file which elements


© 2023 Cadence Design Systems, Inc.

over what time period have current over the threshold for a time period equal to or greater than the specified duration. If no window is specified, the whole simulation period is used.

1. Adjust the power checking analysis settings as needed.
2. Click *Add* to add a power check.

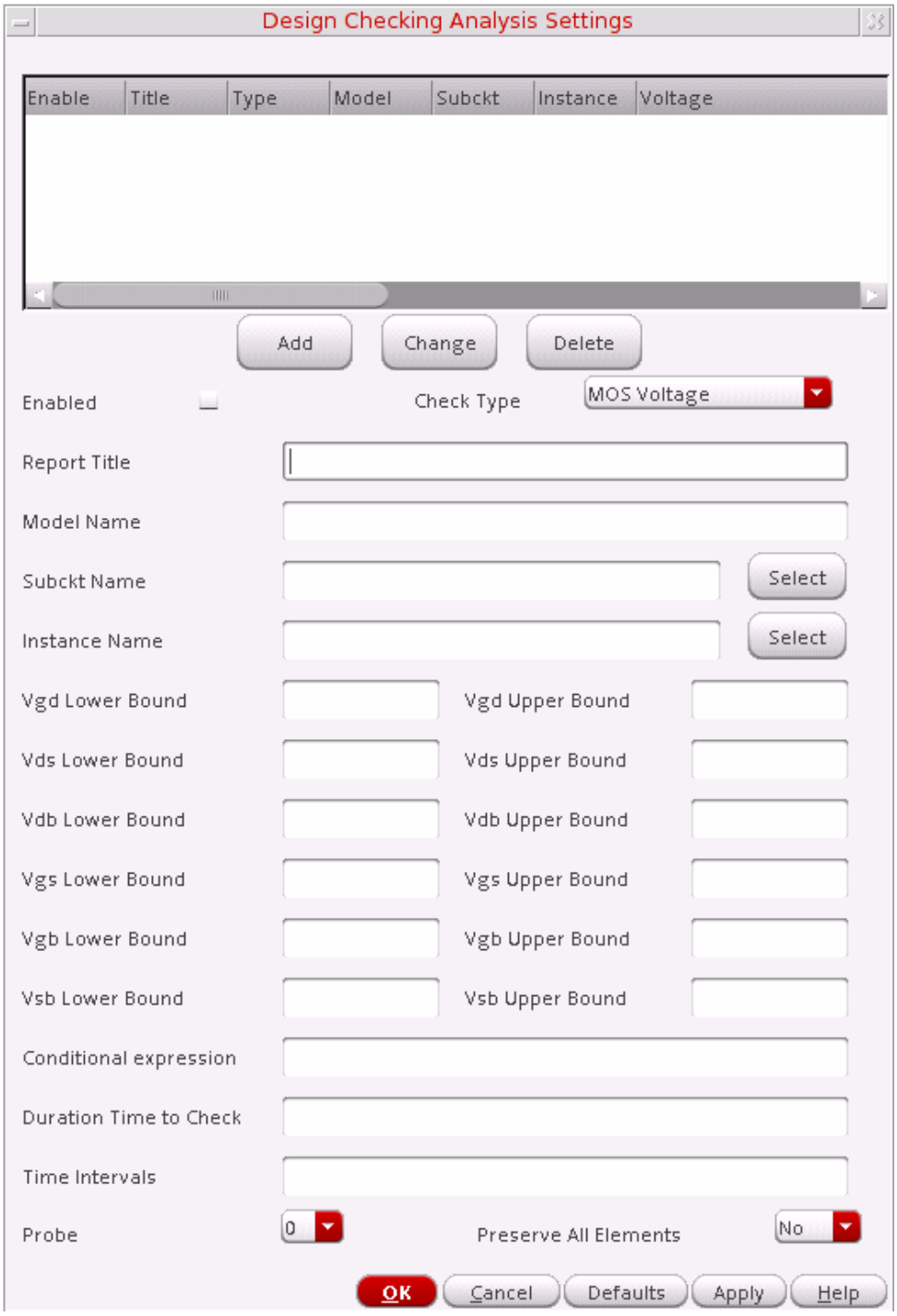
Power checking results can be viewed from the ADE Explorer window using *Results – Print – Advanced Analysis Results – Power Check Report*.

For more information about the power checking analysis settings, refer to *Virtuoso UltraSim Advanced Analysis* chapter in the *Virtuoso UltraSim Simulator User Guide*.


© 2023 Cadence Design Systems, Inc.

## Design Checking Analysis

This command allows you to monitor device voltages during a simulation run, and generates a report if the voltages exceed the specified upper and lower bounds.





© 2023 Cadence Design Systems, Inc.

1. Adjust the design checking analysis settings as needed.
2. Click *Add* to add a design check.

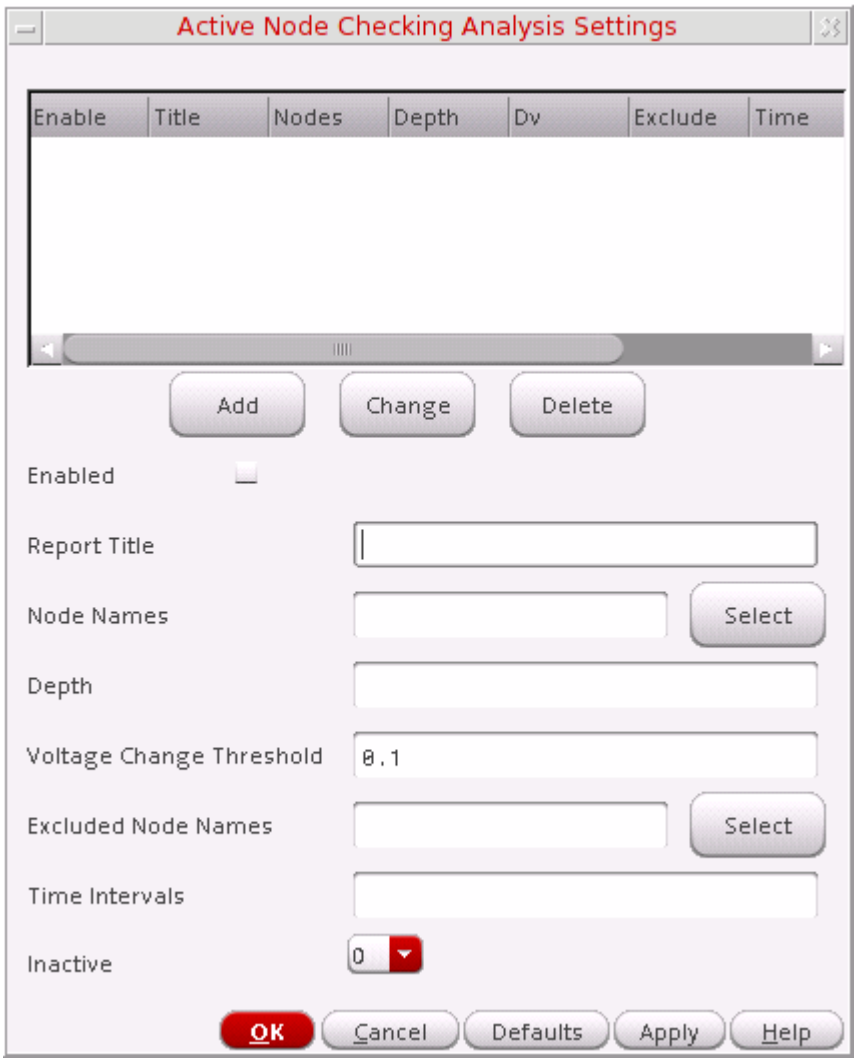
Design Checking results can be viewed from the ADE Explorer window using *Results – Print – Advanced Analysis Results – Device Voltage Check Report*.

For more information about the design checking analysis settings, refer to *Virtuoso UltraSim Advanced Analysis* chapter in the *Virtuoso UltraSim Simulator User Guide*.


© 2023 Cadence Design Systems, Inc.

### Active Node Checking Analysis

Active Node Checking analysis detects nodes with voltage changes that exceed the user defined threshold. With the active nodes identified, you can choose to selectively backannotate parasitic elements during post-layout simulation.




1. Adjust the active node checking analysis settings as needed.
2. Click *Add* to add an active node check.

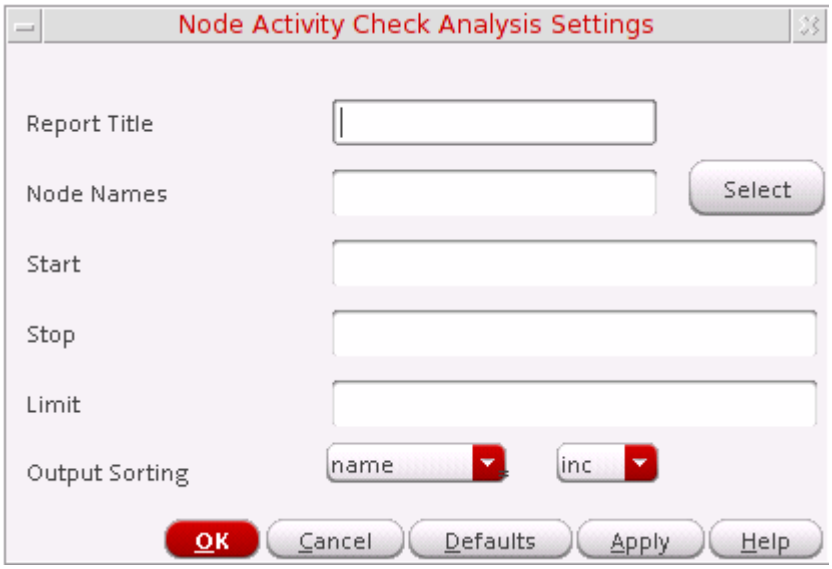
Active node checking results can be viewed from the ADE Explorer window using *Results – Print – Advanced Analysis Results – Active Node Check Report*.

For more information about the active node checking analysis settings, refer to *Virtuoso UltraSim Advanced Analysis* chapter in the *Virtuoso UltraSim Simulator User Guide*.


© 2023 Cadence Design Systems, Inc.

### Node Activity Analysis

Node Activity analysis provides information about the nodes and monitors activities such as voltage overshoots (VOs) and voltage undershoots (VUs), maximum and minimum rise/fall times, signal probability of being high or low, node capacitance, and number of toggles.




1. Enter the node names or click *Select* to choose the same using the schematic.
2. Enter the start and stop time.
3. Adjust the Node Activity Check analysis settings as needed.
4. Click *OK* to add a node activity analysis check.

Node activity analysis results can be viewed from the ADE Explorer window using *Results – Print – Advanced Analysis Results – Node Activity Analysis*.

For more information about the node activity analysis settings, refer to *Virtuoso UltraSim Advanced Analysis* chapter in the *Virtuoso UltraSim Simulator User Guide*.

### Node Connectivity Analysis

The Virtuoso UltraSim simulator helps you perform Node Connectivity analysis using `.usim_report` commands. The information is reported in a `.pr` file. For example, if the netlist name is `circuit.sp`, then the report is named `circuit.pr`.


© 2023 Cadence Design Systems, Inc.

The `.usim_report` commands are useful for debugging simulations.

1. Adjust the node connectivity settings as needed.
2. Click *OK* to add a node connectivity analysis check.

Node Connectivity Analysis results can be viewed from the ADE Explorer window, using *Results – Print – Advanced Analysis Results – Node Connectivity Analysis*.

For more information about the partition and node connectivity analysis settings, refer to *Virtuoso UltraSim Advanced Analysis* chapter in the *Virtuoso UltraSim Simulator User Guide*.

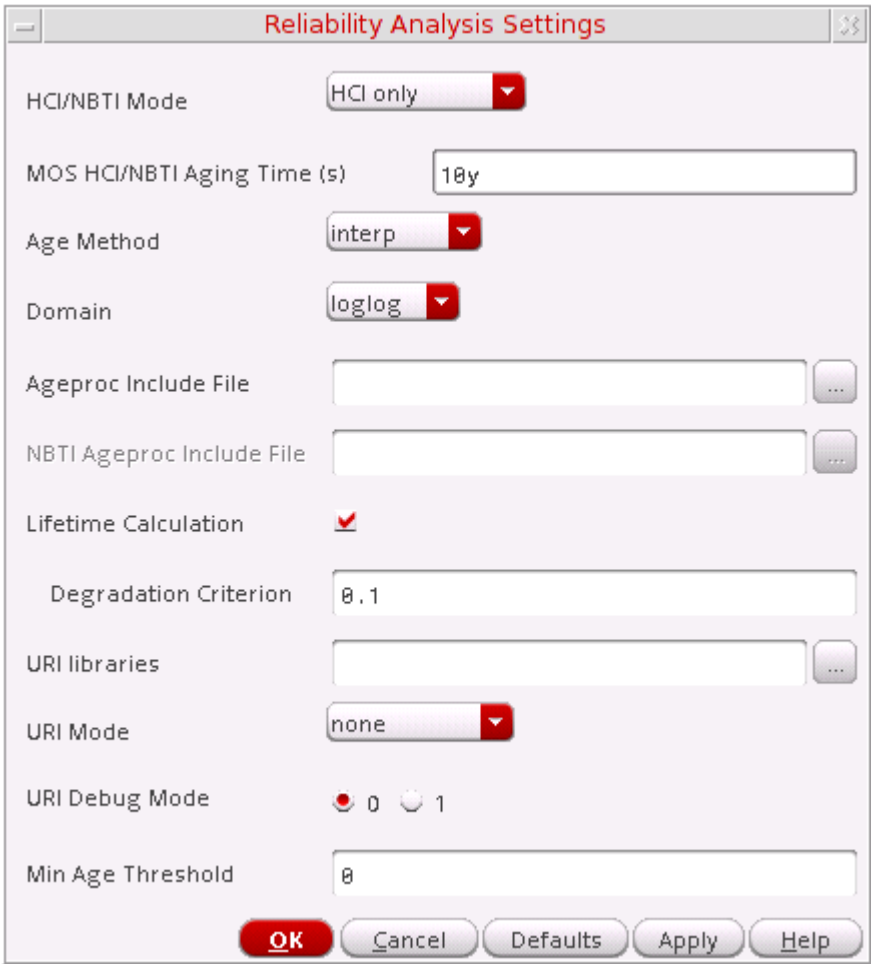
### Reliability Analysis

Reliability analysis simulates circuit aging due to hot carrier injection (HCI) induced degradation and negative bias thermal instability (NBTI). It can also simulates degraded circuit performance for the above effects, after a specified amount of circuit operation.

To run a reliability analysis, you need reliability models which are usually provided by your modeling group. Reliability models can be added using *Setup – Model Libraries* in ADE Explorer.


© 2023 Cadence Design Systems, Inc.

**Note:** Reliability analysis is only available with HSPICE netlist format.




1. Adjust the reliability analysis settings as needed.
2. Click *OK* to add a reliability analysis check.

Reliability analysis results can be viewed from the ADE Explorer window using *Results – Print – Advanced Analysis Results – Reliability Analysis*.

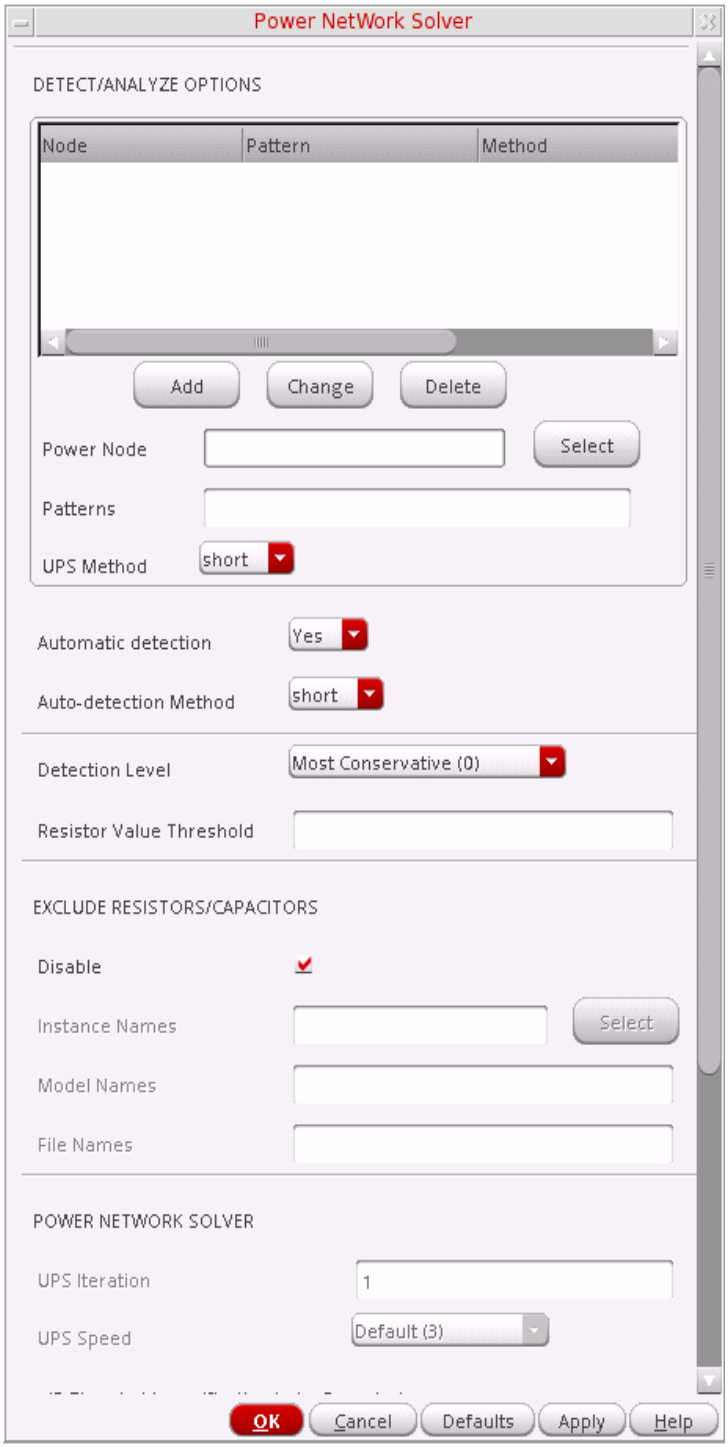
For more information about the reliability activity analysis settings, refer to *Virtuoso UltraSim Advanced Analysis* chapter in the *Virtuoso UltraSim Simulator User Guide*.

### Power Network Solver

Power Network Solver is an optimized solver designed to analyze linear power networks. The solver is integrated into Virtuoso Ultrasim Simulator and together with the Virtuoso Ultrasim


© 2023 Cadence Design Systems, Inc.

engine, helps you calculate the IR drop in power networks and analyze the effects of IR drop on circuit behavior.





© 2023 Cadence Design Systems, Inc.

1. Adjust the Power Network Solver settings as needed.
2. Click *OK* to add a this check.

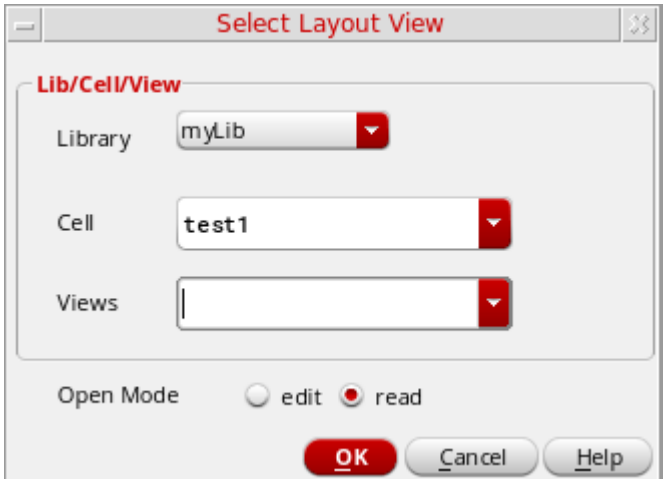
Power Network solver can be viewed from the ADE Explorer window, using *Results – Print – Advanced Analysis Results – IR Drop Report*.

For more information about the Power Network Solver, refer to refer to *Power Network Solver* chapter in the *Virtuoso UltraSim Simulator User Guide*.

### Viewing Output of an EM/IR Analysis

After the EM/IR analysis is run, you can do the following

- To view EM/IR run log, choose *Results – EM/IR Data – EM/IR Log*.
- To view the EM/IR report, choose *Results – EM/IR Data – Report*.
- To view the layout analysis, choose *Results – EM/IR Data – Layout Analysis*. The Select Layout View form appears in which you specify the library, cell, and view information.





© 2023 Cadence Design Systems, Inc.

© 2023 Cadence Design Systems, Inc.

# 4

# Specifying Output Data to be Saved or Plotted

This chapter covers the following topics to explain how to select data that you want to save or plot during a simulation run.

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.

This chapter covers the following topics:


- Using the Outputs Setup Pane on page 406
- Type of Outputs to be Saved or Plotted on page 407
- Defining Specifications for an Output on page 450
- Importing and Exporting Outputs to a CSV File on page 467
- Specifying which Outputs to Save on page 472
- Specifying which Outputs to Save on page 472
- Expressions to be Measured Across Dimensions on page 489
- Environment Variables for PSFXL Output Format on page 490



© 2023 Cadence Design Systems, Inc.

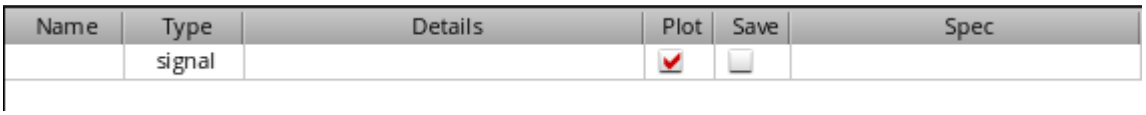
# Using the Outputs Setup Pane

The Outputs Setup pane appears to the right of the ADE Explorer window. You can use the this pane to add a new output signal/expression/OCEAN or MATLAB script or to change an existing output. This pane is blank when no outputs are specified.

Do the following to add a new output:

- Choose *Outputs – Add –* <Type of output>.
- On the Run toolbar, click the button. The list of outputs appears. Select the type of output you want to add.

The selected output type is added in a new row in the Outputs Setup pane. For example, if you add an output of type signal, the following row appears in the Outputs Setup pane. The Plot check box is selected by default.




## Specifying Whether a Result is to 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 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>
...
```

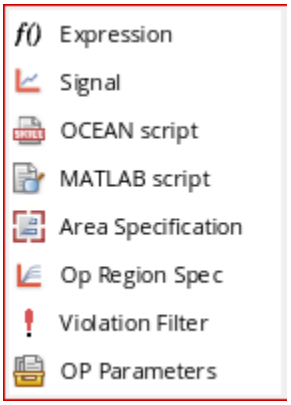
When the *Plot* and *Save* checkboxes are disabled for an output, the *Value* column for that output is blank.

When *Plot* is disabled and *Save* is enabled, a waveform icon is displayed in the *Value* column to indicate that the data is available and can be manually plotted. In this case, the signal is not plotted automatically after the simulation run is complete.


© 2023 Cadence Design Systems, Inc.

# Type of Outputs to be Saved or Plotted

The outputs of type—Expression, Signal, OCEAN Script, MATLAB Script, Area Specification, Violation Filter and OP Parameters—that are added in the Output Setup pane are called measurements. The following figure shows the list of outputs that can be saved or plotted after a simulation is complete:



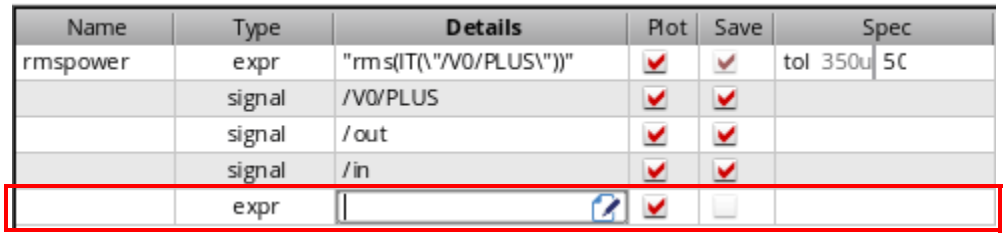

This section covers the following topics:

- Saving Output Expression on page 408
- Saving Output Signals on page 414
- Saving OCEAN or MATLAB Measurement on page 418
- Defining Specifications for an Output on page 450
- Saving Violation Filter on page 453
- Saving Operating Point Parameters on page 453
- Adding a User-Defined Column on page 460
- Renaming a User-Defined Column on page 461
- Deleting a User-Defined Column on page 461


© 2023 Cadence Design Systems, Inc.

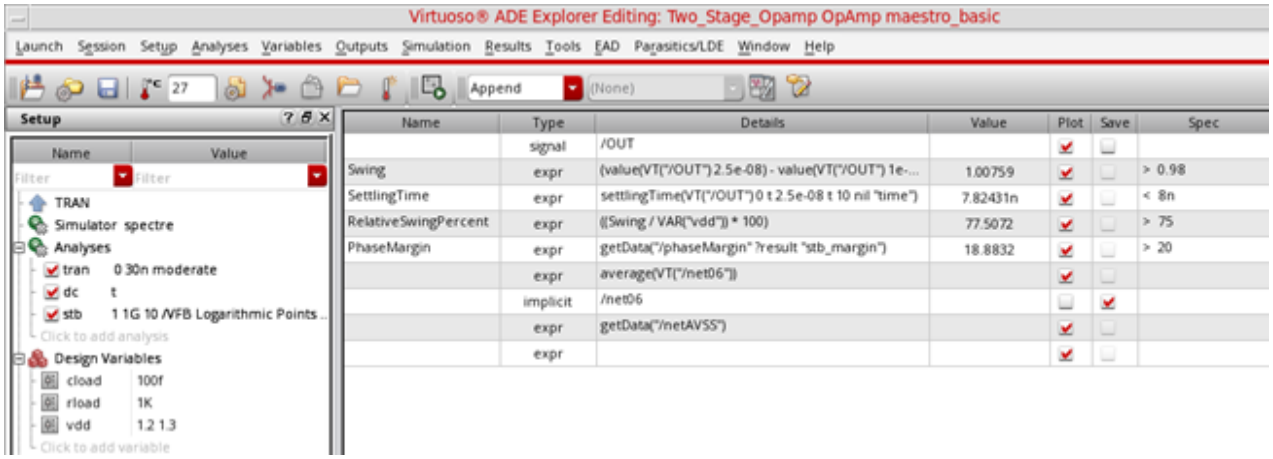
## Saving Output Expression

To add output expressions, click *Expression* from the drop-down list. A new row of type `expr` is added in the Outputs Setup tab.




1. In the *Name* column, specify the name for the expression output.
2. In the *Details* column, either directly type an output expression or click the button to open Expression Builder to add 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")`. For more information about how to build expressions using Expression Builder, see Creating Expression Outputs Using the Expression Builder.
3. The expression appears in the *Details* field. If the expression uses a signal or net that does not already exist in the outputs table, a new row for an output of type `Implicit` is added to the table. By default, implicit signals are not shown in the outputs table. To show these signals, set the environment variable displayImplicitSignals to `t`

   As shown in the example below, when you create `average(VT("/net06"))` as an output expression, an implicit output is created for signal `/net06`.





© 2023 Cadence Design Systems, Inc.

*Important*

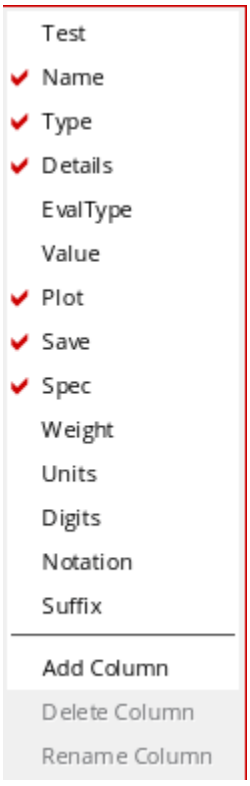
In the following two scenarios, the implicit signals are not added to the output setup:

- The output expressions, containing implicit signals, were created using ADE Explorer or Assembler (IC6.1.8 ISR5 version or below). The simulation does not consider these expressions.
- The *Select signals to output (save)* option on the Save options form is set to *All*.

To add the implicit signals to the outputs setup, set the *Select signals to output (save)* option on the Save options form to *selected*.

When the netlist is being created in the GUI, i.e. the ICRP mode, you can set the saveImplicitSignalsDuringRun environment variable to `t`. This ensures that the implicit signals are added to the `save` statement in the netlist during the simulation run. However, those implicit signals will not be visible in the *Outputs Setup* or *Results* tab.

1. Change the *Plot* and *Save* check boxes for the expression, if required.
2. To add more columns to the Outputs Setup pane for expression, right-click anywhere in the header and choose the column you want to add from the context-menu. By default, *Name*, *Type*, *Details*, *Plot*, *Save*, and *Spec* columns are displayed.





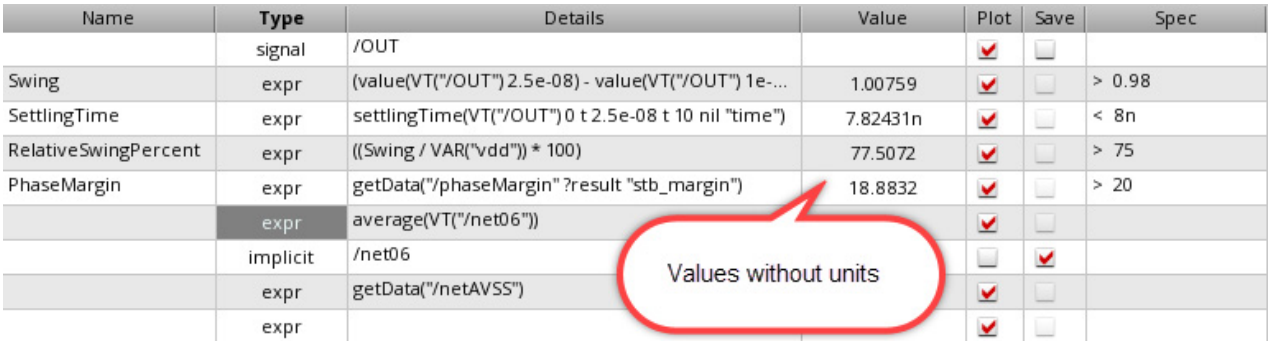
© 2023 Cadence Design Systems, Inc.

3. When you select to display the *EvalType* column, the default value displayed in this column is `point`. This signifies that by default the expression are run for the given design point. If you want to evaluate the expression across corners, or across sweep points, or across all corners and sweep points, double-click in the *EvalType* cell and select `corners`, `sweep`, or `all` from the drop-down list.

   When you set the *EvalType* to a value other than point, the color of the row is changed to blue.

4. In the *Value* column, the value of each output type is displayed. The default value in this column is a blank string.

   **Note:** The environment variable showUnitsInResults is used to control whether to show or hide the units, such as `Hz` or `dB`, in the measured results or values for the expressions in the Results tab.




5. In the *Spec* and *Weight* columns next to the expression, define the specification and the weighting factor for the specification for the output expression.

6. 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.

7. 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.


© 2023 Cadence Design Systems, Inc.

8. 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. |
| *eng* | Display results in the engineering notation. |
| *sci* | Display results in the scientific notation. |
| *suffix* | Display results in the suffix notation.<br><br>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.

**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 versus Floating-Point Division in the Arithmetic and Logical Expressions chapter of the *Cadence SKILL Language User Guide*.

### Creating Dependent Expressions

You can create expressions based on other expressions. For example, assume that you have an expression named `s1` with the value `5`. If you want another expression, say, `s2` to be ten times the value of `s1`, type `s1*10` in the *Details* field as shown in the figure below.

| Name | Type | Details | EvalType | Plot | Save | Spec |
|---|---|---|---|---|---|---|
| S1 | expr | 5 | point | ☑ | ☐ | |
| S2 | expr | (S1 * 10) | point | ☑ | ☐ | |


© 2023 Cadence Design Systems, Inc.

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.

  **Note:** 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` and then use `myVout2Expr` in a dependent expression.

- Error messages, if any, regarding dependent expressions are displayed in the Command Interpreter Window (CIW).

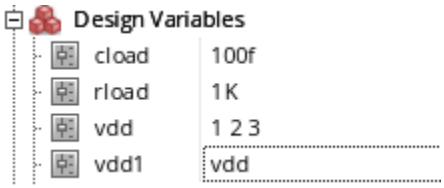
Dependent expressions appear on the *Results* tab only if the *Plot* check box is checked. However, if you want to display of the results of intermediate expressions irrespective of the setting of the *Plot* check box, set the `forceShowAutomaticExpressions` environment variable to `t`.


© 2023 Cadence Design Systems, Inc.

### *Plotting Dependent Expressions*

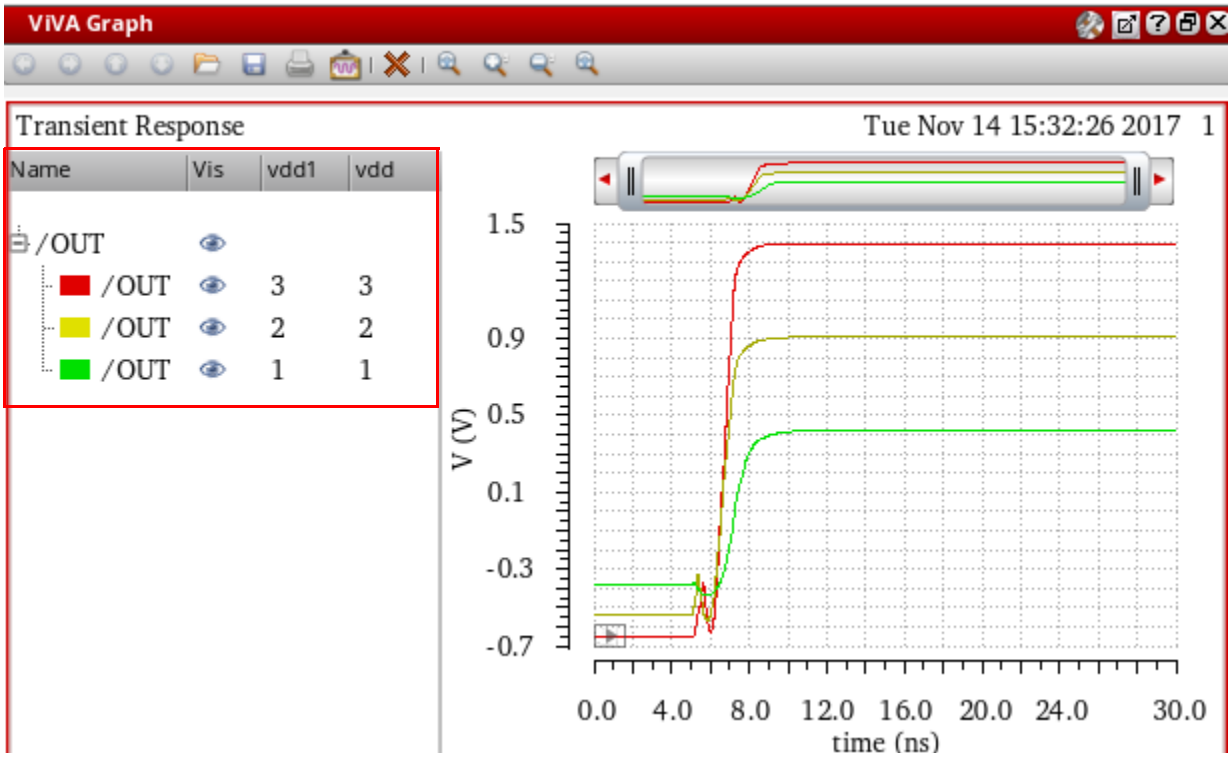
When sweep includes dependent design variables, scalar expression outputs are plotted as continuous line.

For example, consider the ADE Explorer setup in which you have the following design variables:




Here, the variable `vdd1` is dependent on another variable `vdd`.

Now, if you run a simulation and plot the `/OUT` signal, the following waveforms are generated for the three values of `vdd` and `vdd1`, as shown in the figure below.





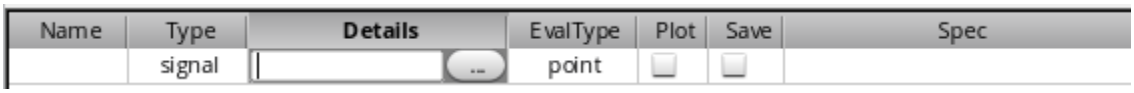
© 2023 Cadence Design Systems, Inc.

## Saving Output Signals

You can selects signals (nodes, terminals, or nets) on the schematic and specify whether you want to save or plot them.

To select an output on the schematic, do the following:

1. Choose *Signal* from the drop-down list. A new row is added with the output type *signal*.




2. In the *Name* field, type the name for the signal output. This name appears in the ViVA XL window.
3. Double-click in the *Details* column and click the ellipses button.

   The schematic window appears in a new tab.
4. In the schematic, select a signal.

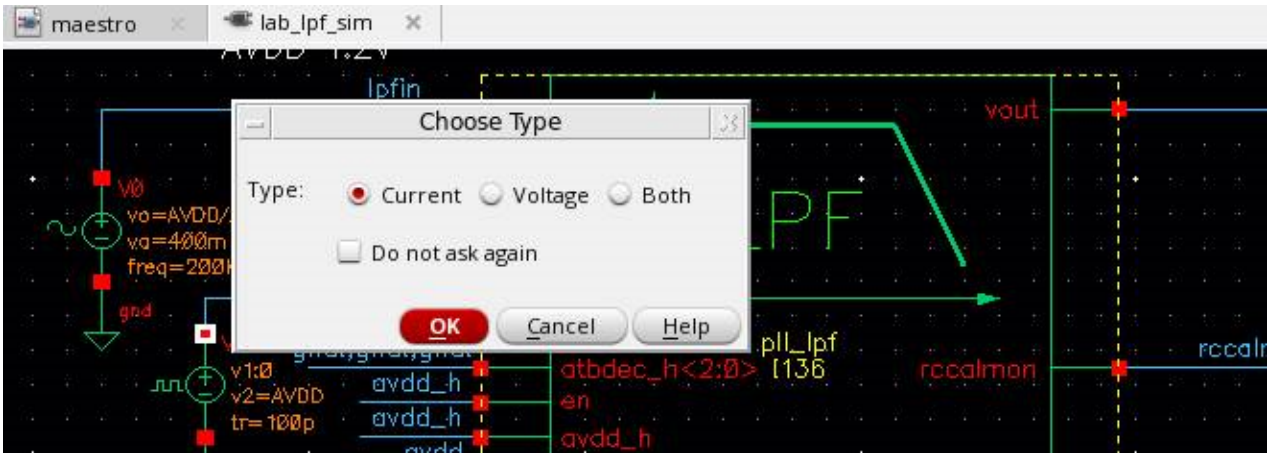
   The signal name appears in the *Details* field on the *Outputs Setup* tab.

   Alternatively, you can manually enter a signal. In this case, the type of data to be saved—current, voltage, or both— is decided through the terminal selection type specified in the *Plotting/Printing Options* form.

   **Note:** By default, the current data is saved for all selected terminals. However, you can choose to save or plot voltage data, or both current and voltage. Alternatively, you can request for a prompt to check which type of data to be saved for all signal output selections. For this, set the terminalSelectionType environment variable.


© 2023 Cadence Design Systems, Inc.

When the variable is set to "prompt", the following message prompt is displayed where you can select the type of data.




5. 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.

Alternatively, in the ADE Explorer window, choose *Outputs – To be Saved – Select On Design* or choose *Outputs – To be Plotted – Select On Design* to save or plot the signals. The schematic design appears. Select the signals you want to save. The selected signals appear in the *Details* column.

You can select multiple signals in schematic. Press Esc to end the selection. The name of each selected item appears (such as */out* or */I2/PLUS*) in different rows in the Outputs Setup tab.

### 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 an instance to choose all instance terminals.
- Click the square pin symbols to choose currents.
- Click wires to choose voltages.


© 2023 Cadence Design Systems, Inc.

- 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.

## Saving a Digital Bus

In the *Outputs Setup* pane, pane, you can specify a digital bus output to be saved or plotted when the simulation run is complete. The digital bus output can include a single bit or multiple bits bundled together in one output.

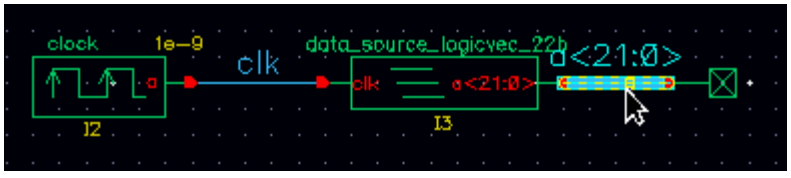
To add a single bit or a bundle of bits from a bus as outputs in the ADE Explorer *Outputs Setup* pane, perform the following steps:

1. Add a new output row of type `signal`. Then, double-click in the *Details* column and click the ellipses button.

   Alternatively, choose *Outputs — To be Saved — Select On Design*, or choose *Outputs —To be Plotted — Select On Design*.

   The schematic appears in a new tab.

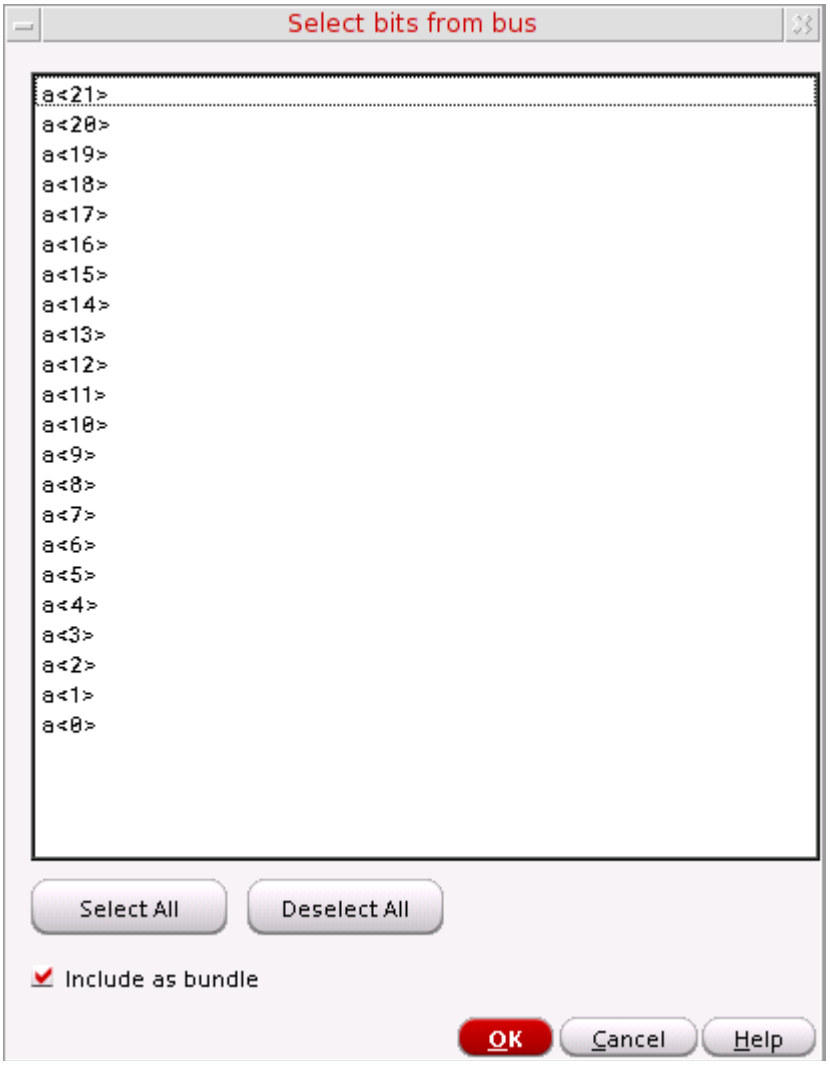
2. In the schematic, click the bus.





© 2023 Cadence Design Systems, Inc.

The Select bits from bus form appears.




Perform one of the following steps to specify bits from the bus as outputs to be saved or plotted:

- To add a single bit as an output, select the bit and click *OK*.
- To add multiple bits bundled together as an output,
  - Press the `Ctrl` key and select the bits.
  - Ensure that the *Include as bundle* check box is selected and click *OK*.

  The selected bits appear in one row in the *Outputs Setup* pane.

  **Note:** If the *Include as bundle* check box is not selected while you are adding or


© 2023 Cadence Design Systems, Inc.

editing an output, an error message is displayed.

- To add multiple bits to individual outputs, use the *Outputs — To Be Saved — Select On Design,* or the *Outputs — To Be Plotted — Select On Design* command in the ADE Explorer window to open the schematic.

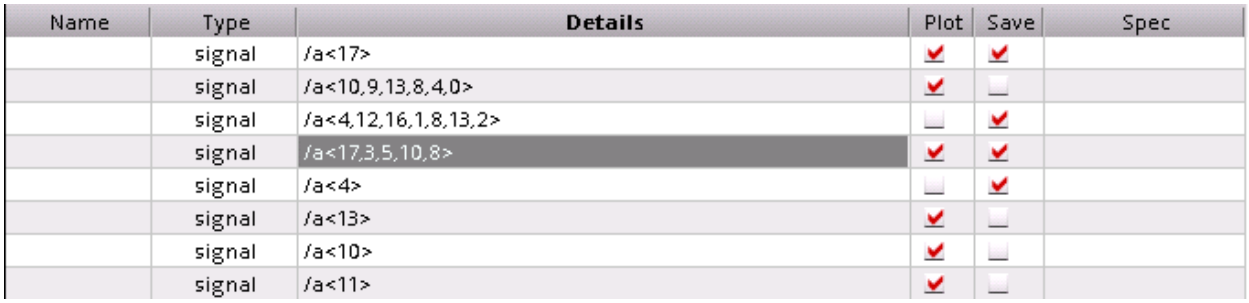
  In the schematic, click the bus to open the Select bits from bus form and perform the following steps:

  - Press the `Ctrl` key and select the bits.
  - Clear the *Include as bundle* check box and click *OK*.

  Each selected bit appears in a different row in the *Outputs Setup* pane.

  You can also click the *Select All* button to select all the available bits. Or use the *Deselect All* button to clear the selection.

The following image shows how the selected bits appear as ouputs in the *Outputs Setup* pane in ADE Explorer.



| Name | Type | Details | Plot | Save | Spec |
|---|---|---|---|---|---|
| | signal | /a<17> | ☑ | ☑ | |
| | signal | /a<10,9,13,8,4,0> | ☑ | ☐ | |
| | signal | /a<4,12,16,1,8,13,2> | ☐ | ☑ | |
| | signal | /a<17,3,5,10,8> | ☑ | ☑ | |
| | signal | /a<4> | ☐ | ☑ | |
| | signal | /a<13> | ☑ | ☐ | |
| | signal | /a<10> | ☑ | ☐ | |
| | signal | /a<11> | ☑ | ☐ | |

3. In the *Name* field on the *Outputs Setup* pane, type the name for the signal output. This name appears in the graph window.
4. You can click or clear the *Plot* or *Save* check box corresponding to the output as required.
5. If you have used the *Outputs — To Be Saved — Select On Design* command, or the *Outputs — To Be Plotted — Select On Design* command to open the schematic, press the `Esc` key to end the selection.

## Saving OCEAN or MATLAB Measurement

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), do the following:

1. Select the OCEAN Script or MATLAB Script in the drop-down list.


© 2023 Cadence Design Systems, Inc.

A new row is added for the test with the output type *ocean* or *matlab*.

See the *OCEAN Reference* for information about OCEAN script commands.

1. In the *Name* column, type a name to represent the measure.
2. In the *Details* column, type the name of (and location/path to) the script file. Alternatively, double-click the *Details* field. The ellipses button appears. Click this button to open the Choose a File form in which you can browse to specify a file. After you select a file and click *Open*, the relative path to the file appears in the *Details* field.

   **Note:** Once the script filename appears in this field, you can edit it by clicking the Edit File button.


© 2023 Cadence Design Systems, Inc.

## 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 the OCEAN or MATLAB script file in the *Details* field.
2. Click the *Edit File* button.

   The script file appears in a text editor.

   **Note:** By default, ADE Explorer opens the text editor set by using the `$EDITOR` shell variable in UNIX. You can specify a preferred text editor by using the `editor` SKILL variable.
3. Edit the script file and save your changes.

   The program uses the new script the next time you run the test.


© 2023 Cadence Design Systems, Inc.

## 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 Explorer results path. If you want your script to write results to the Results tab, 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 Explorer 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, 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`.


© 2023 Cadence Design Systems, Inc.

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.

```
out = cds_srr( axlCurrentResultsPath,'tran-tran', 'out' )
fig = figure
plot( out.V,out.time )
axlResult = max( out.time )
```


© 2023 Cadence Design Systems, Inc.

## Creating Expression Outputs Using the Expression Builder

You can use the Expression Builder to create outputs of type `expr` in the ADE Explorer Outputs Setup pane that are evaluated after a simulation run is complete. The Expression Builder provides an interface that helps you create or edit expressions based on suggested inputs, with minimal interaction between the schematic and Calculator. However, using the Expression Builder, you can open the function template, and also select nets and instances from the schematic.

This section includes the following topics:

- Opening the Expression Builder
- Graphical User Interface of Expression Builder
- Supported Functions
- Creating Expressions Using the Expression Builder
- Using Schematic-Selection Functions
- Using the Existing Function Templates
- Using Function Templates to Specify Input Signals
- Using Functions with a Signal Argument
- Using VAR in Expressions
- Creating Nested Expressions
- Creating an Expression using CalcVal
- Bindkeys in Expression Builder
- Evaluating or Plotting Expressions in Expression Builder

Video

To know more about how to use eye masks, view the Expression Builder in ADE Assembler and Explorer video available on Cadence Online Support.

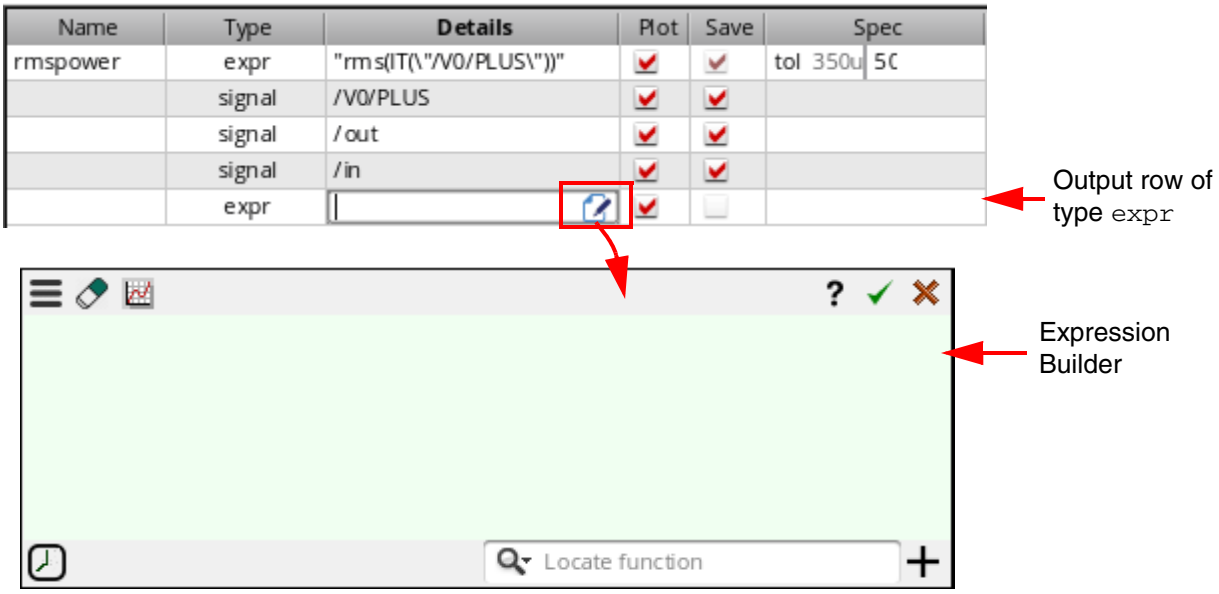
In addition, you can read an interesting blog about Expression Builder, see Virtuoso Video Diary: Why Should you Switch to the Expression Builder for Creating Expressions?.


© 2023 Cadence Design Systems, Inc.

### Opening the Expression Builder

Perform the following steps to open the Expression Builder:

1. In the Outputs Setup pane, add an output row of type `expr`.
2. Double-click the *Details* column of this signal row. An icon appears at the end of this column. Click this icon to open the Expression Builder, as shown in the figure below:




Now, you can use Expression Builder to create the expressions.

**Note:** You can also edit an existing output expression using the Expression Builder. To edit an existing expression, double-click in the *Details* column for an existing output row. Then, click the Open Expression Builder button. The selected expression is displayed in the Expression Builder that you can edit further.

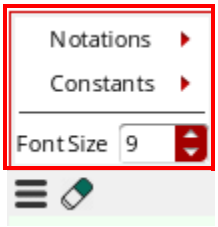
### Graphical User Interface of Expression Builder

The Expression Builder GUI includes the following elements:

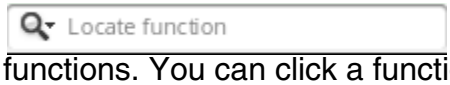
- Top Bar—Includes the following buttons:


© 2023 Cadence Design Systems, Inc.

- Menu—When you click the Menu button the following options appear:




  - *Notations*—Includes a list of notations that you can use to represent data in the expression. Available options are: `Engineering`, `Suffix`, and `Scientific`. The default value is `Engineering`.
  - *Constants*—Includes a list of constants, such as `boltzmann`, `charge`, `degPerRad`, `epp0`, `pi`, `twoPi`, `sqrt2`. When you select a constant, it is added to the current cursor position in the Buffer.
  - *Font Size*—Specifies the font size of the expression displayed in the Buffer.
- —Clears the data in the Buffer.
- —Displays the evaluation and plot options. For more information, see Evaluating or Plotting Expressions in Expression Builder on page 447.
- —Opens the related documentation.
- —Adds the expression displayed in the Buffer to the Outputs Setup pane.
- —Cancels and closes the Expression Builder.

- Buffer—An input area where the expressions you build are displayed.
- Bottom Bar—Includes the following buttons:
  - *Recent* drop-down list—Displays a list of all the expressions that have been created in the Expression Builder. This drop-down list works like a stack. When you click a listed expression, it is added to the cursor position in the Buffer. The up and down arrow keys and tab can be used to navigate through the list and press the `Enter` key to add the selected item to the Buffer of the Expression Builder.
  - 

—Displays a list of all the available Calculator functions. You can click a function to open its template, then specify the argument values, and click *OK* to add it to the Buffer.
  - — Opens the Add Custom Calculator Function form that you can use to create custom functions. For more information about how to create custom functions, see


© 2023 Cadence Design Systems, Inc.

Creating Custom Calculator Functions in *Virtuoso Visualization and Analysis XL User Guide*.

### Supported Functions

- Calculator Functions—The following Calculator function categories are supported in the Expression Builder:
  - Special Functions
  - Modifier Functions
  - Trigonometric Functions
  - Math Functions
  - RF Functions
  - Spectre RF Functions
- Schematic Selection Functions—These are a set of standard functions that can be applied on the schematic objects to build expressions. When you click a function button on the toolbar, the schematic design view appears. In the schematic design view, select a schematic object on which you want to apply the selected function. When you select the schematic object, an expression is created and displayed in the Buffer.
- CalcVal—This function can be used to return the value of an output expression specified in the same or another test. The value returned by this function can be used later in another expression.
- Custom Functions—You can create your own functions (custom functions) and add them in the Function Panel. The existing and newly-added custom functions appear in blue in the Locate function list. In addition, custom functions appear in the Autocomplete drop-down list.

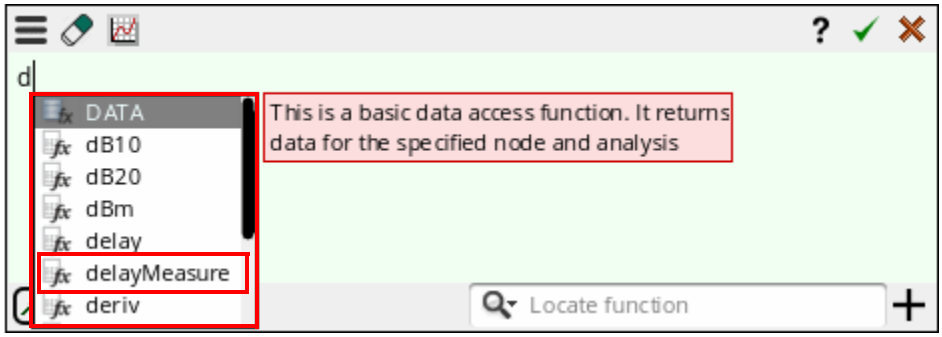
### Creating Expressions Using the Expression Builder

To create expressions in the Expression Builder:

1. Type the expression in the Buffer. As you start typing, a drop-down list appears with suggested values. For example, if you type letter `d`, all the functions available in the Calculator starting with letter `d` are listed in the drop-down list that you can apply in the

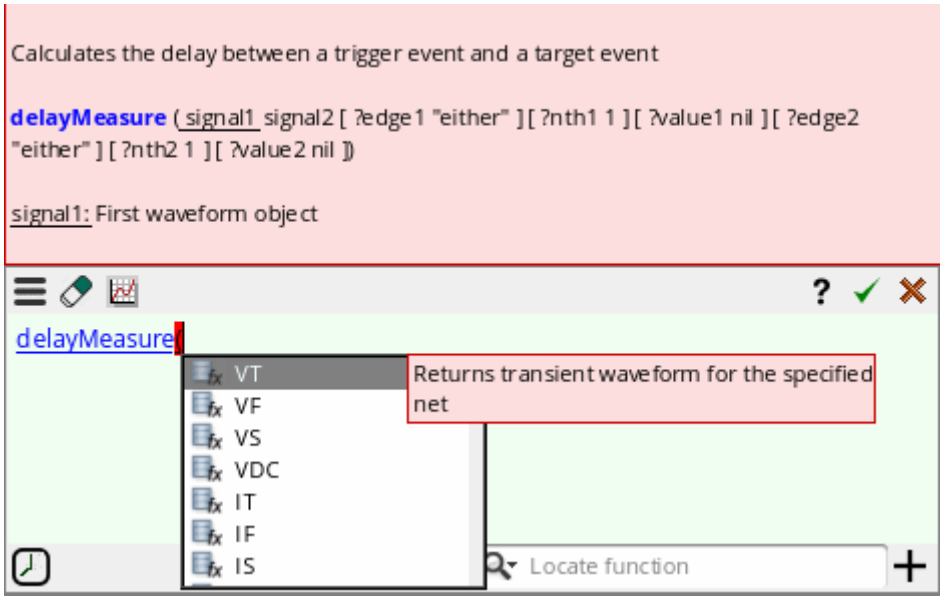

© 2023 Cadence Design Systems, Inc.

current scenario. (See the figure below). In addition, when you hover the pointer on a function, a brief description about the selected function is displayed.




2. Now, when you choose a function from the list, the syntax of the selected function appears. For example, if you select the `delayMeasure` function from the given list, the syntax for `delayMeasure` appears in the tooltip that guides you through the arguments required for the selected function (see the figure below).

3. After you type the function name, the next possible input value for this expression is listed in the Auto-complete drop-down list. You can scroll down in the drop-down list using the up and down keyboard arrows and press the `tab` key to select the function. When the function name is complete, the parenthesis will open automatically. For example, see the figure below in which the `delayMeasure` function is selected from the drop-down list.



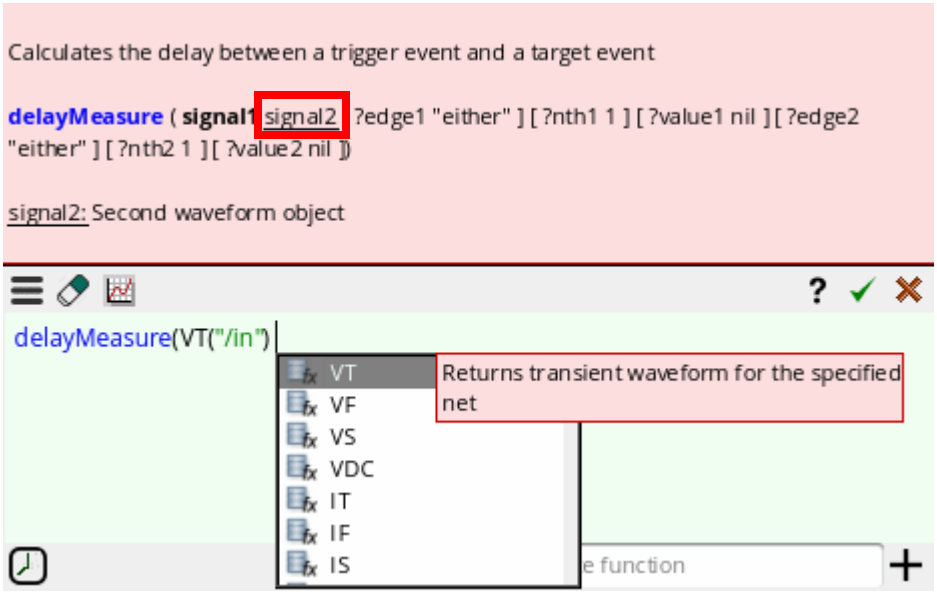

4. To create the expression for the `delayMeasure` function, you need to specify an input signal. After you select the function name, `delayMeasure`, the Auto-complete drop-down list displays a list of schematic-selection functions, such as `VT`, `VF`, and so on. You can select one of these functions to specify the input signal. For more information about


© 2023 Cadence Design Systems, Inc.

how to use these functions to select signals, see Using Schematic-Selection Functions on page 431. Notice that the function tooltip also indicates that the current input argument is `signal1` and displays it with an underline.

**Note:** The Auto-complete drop-down list also includes signals that are already saved in the Outputs Setup pane. You can directly select them from the drop-down list, without the need to open the schematic design.

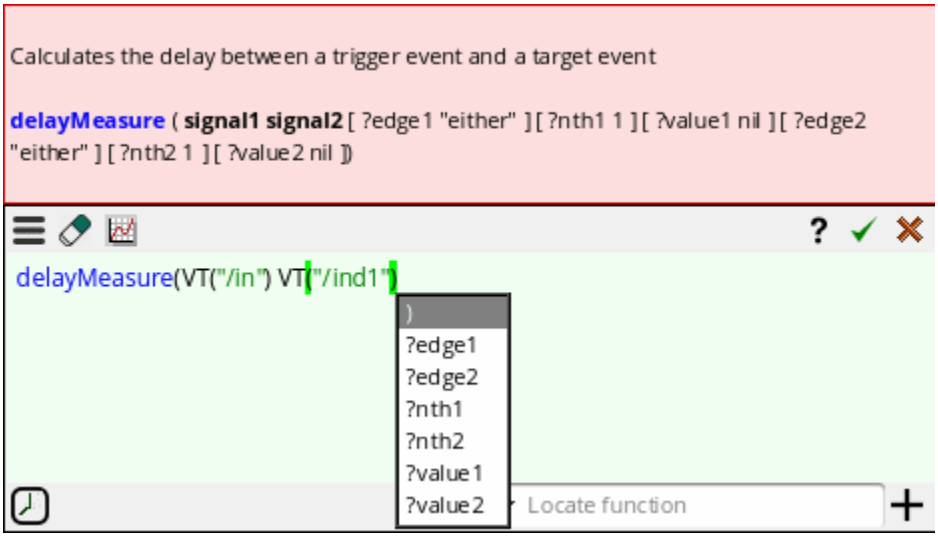
5. After you select the first signal, the tooltip shifts the underline over `signal2`, which is the next argument. In addition, the Auto-complete drop-down list also shows the schematic-selection functions to select the second signal.



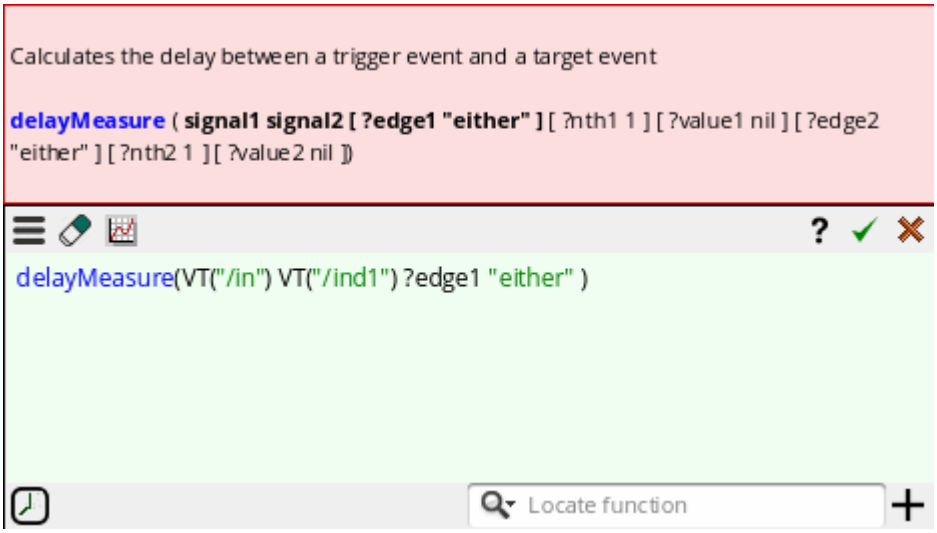

6. After you select `signal2`, the Auto-complete drop-down list displays a list of all the key arguments (optional) starting with `?`, as shown in the figure below. Notice that the tooltip


© 2023 Cadence Design Systems, Inc.

also shows the syntax of these arguments. You can specify these arguments if you want; otherwise, choose the closing parenthesis to complete the expression.




Notice that all the function arguments that you have specified in the expression are displayed in bold in the tooltip.

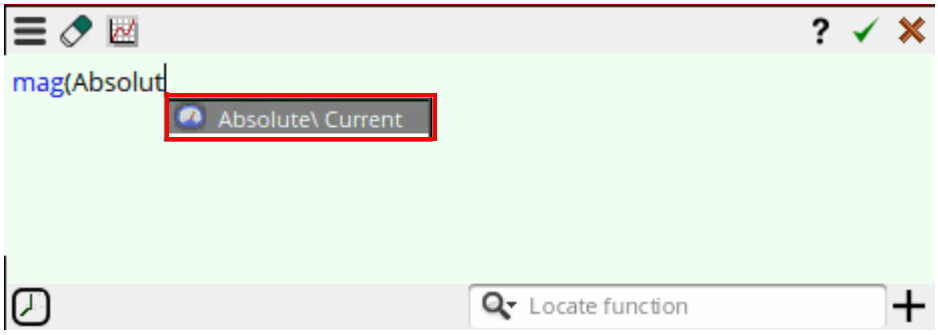



7. After the expression is complete, press the `Enter` key or click the *OK* button available on the Expression Builder.

**Note:** You can also right-click the output row in ADE Explorer or ADE Assembler and choose *Send to Calculator* to open the Calculator window with the selected expression displayed in the buffer.


© 2023 Cadence Design Systems, Inc.

If an expression output name includes a space, such as `Absolue Current`, then the Auto-complete drop-down list displays an escape character (`\`) in this output name, as shown in the figure below:




After you create the expression, the Outputs Setup pane also displays escape character in the expression.

| Name | Type | Details | Value |
|---|---|---|---|
| | signal | /V1/PLUS | |
| | signal | /OUT | |
| Absolute Current | expr | abs(IDC("/V1/PLUS")) | 1.126m |
| UGF | expr | cross(dB20(mag(VF("/OUT")))) 0 1... | 464.2M |
| Gain | expr | value(dB20(mag(VF("/OUT")))) 1) | 46.74 |
| Voffset | expr | (VDC("/inm") - VDC("/inp")) | 3.065m |
| | expr | mag(Absolute\ Current) | 1.126m |

You can use `Alt + Enter` to enter line breaks while creating an expression in the expression builder. This improves readability, and makes it easy to edit long expressions.

These line breaks in the expression are retained even when you close and reopen Expression Builder. Line breaks in the expression are retained also when outputs are exported to the CSV file. When you import these outputs using the CSV file again, line breaks in the expression remain as they were earlier

### *Different Icons Available in Auto-complete Drop-Down List*

The following icons are used to display different types of data in the Auto-complete drop-down list:

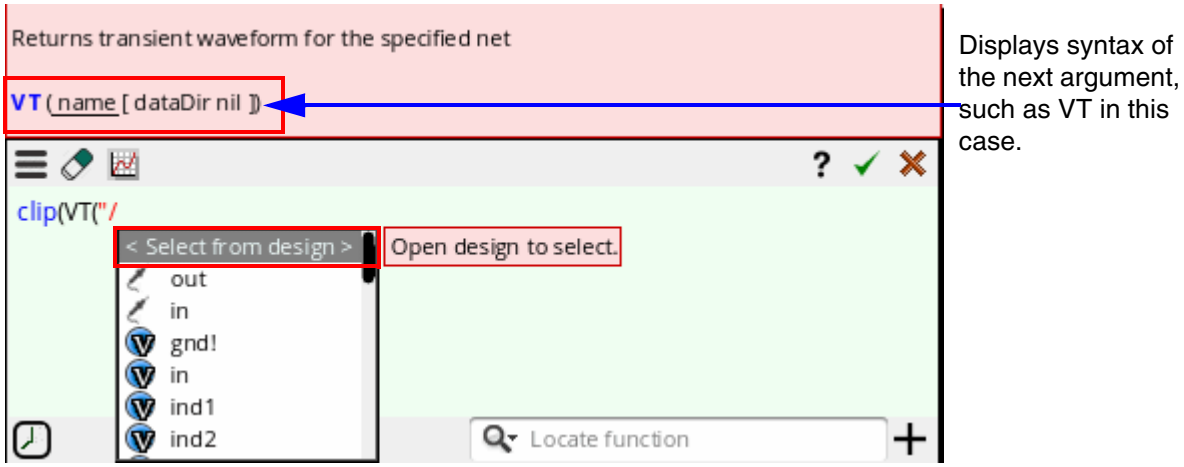
- gnd! — Indicates a voltage signal.
- A — Indicates a current signal.
- V0/PLUS — Indicates signals already available in the Outputs Setup pane.


© 2023 Cadence Design Systems, Inc.

- C7 — Indicates a hierarchical node in the design.
- "CAP" — Indicates a design variable.
- "rmspower" — Indicates an output expression available in the Outputs Setup pane.
- abs_jitter — Indicates a Calculator function or the schematic-selection functions available in the Calculator.

### Using Schematic-Selection Functions

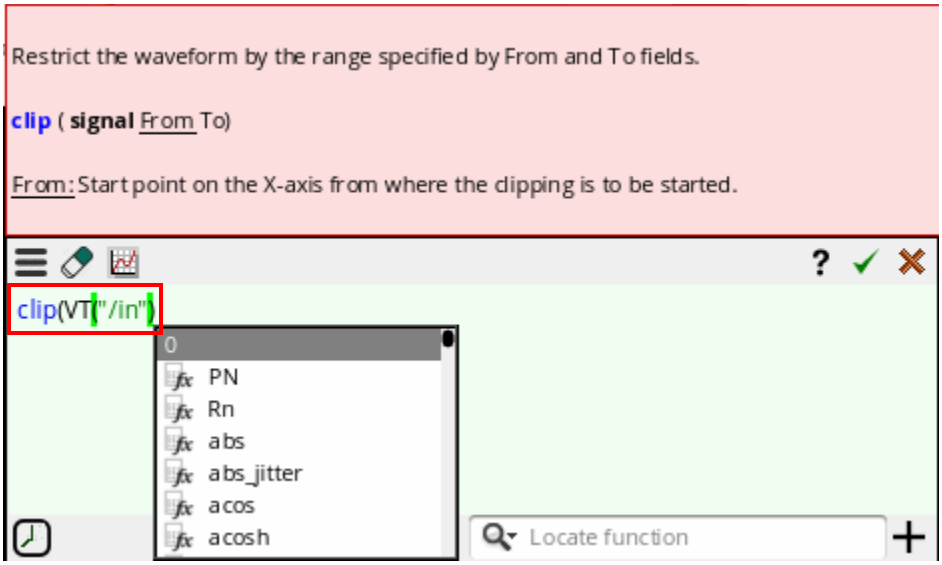
If an argument in the function syntax requires a signal to be chosen from the schematic, then the schematic-selection functions, such as `VT`, are also listed in the Auto-complete drop-down list. After you select the schematic-selection function, the Auto-complete drop-down list then displays the list of available nets and also provides the option, `Select from design`, to choose the net, instance, or terminal by selecting them in the schematic. For example, to create the expression for the `clip` function, you can use the schematic-selection function as shown in the figure below:





© 2023 Cadence Design Systems, Inc.

When you choose this option, the schematic design appears. After you make a selection, the selected object is added in the buffer of the Expression Builder, as shown in the figure below:

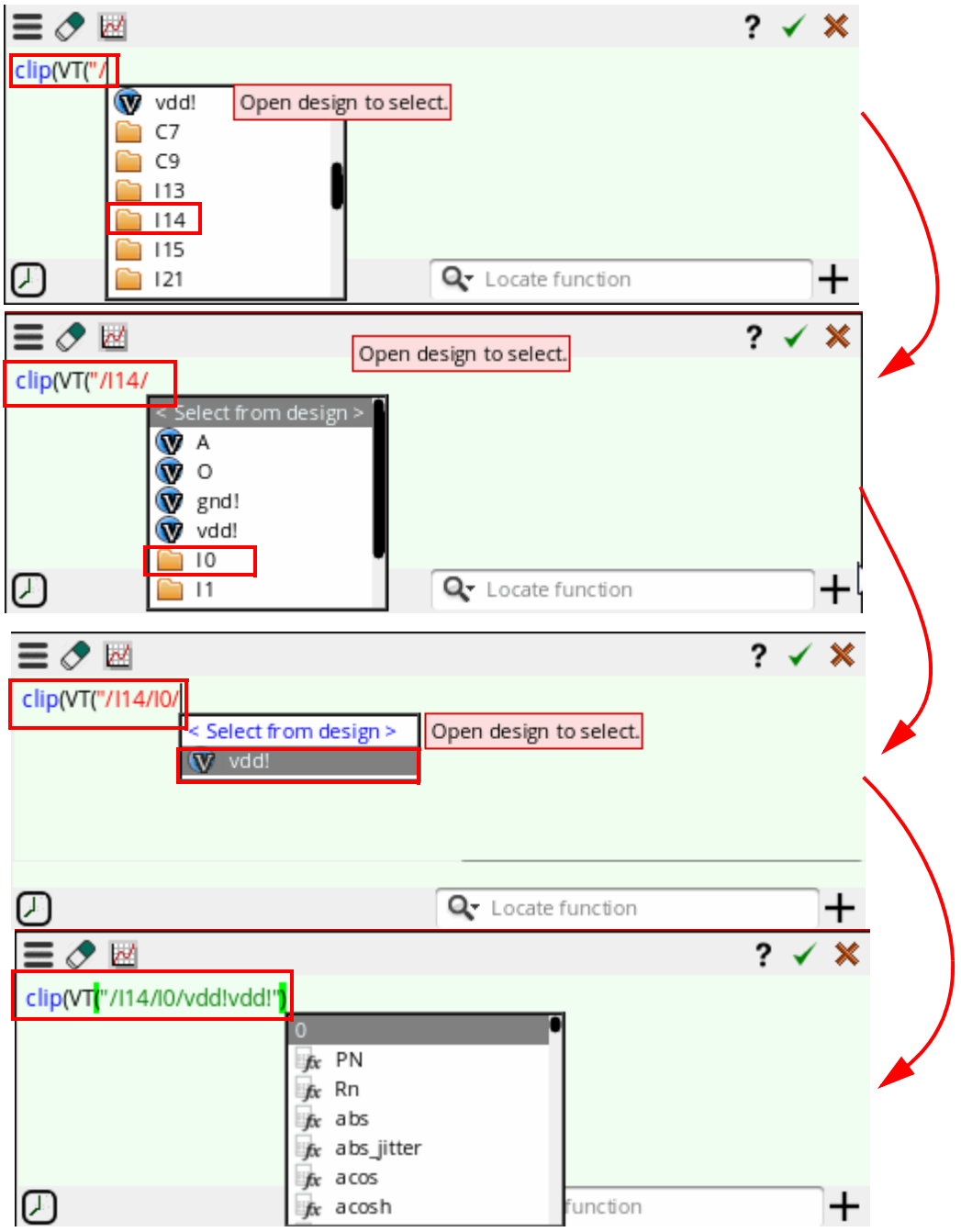



The suggestions for the next argument values are also displayed in the Auto-complete drop-down list. You can select the next argument value based on the syntax given in the tooltip and complete the expression.

After the expression is complete, press the `Enter` key or click the *OK* button available on the Expression Builder.


© 2023 Cadence Design Systems, Inc.

Alternatively, you can type the design elements in the Buffer. After typing the schematic-selection function `VT`, if you select the first-level design element from the drop-down list, the design is traversed and the drop-down list then displays the next hierarchy-level elements.



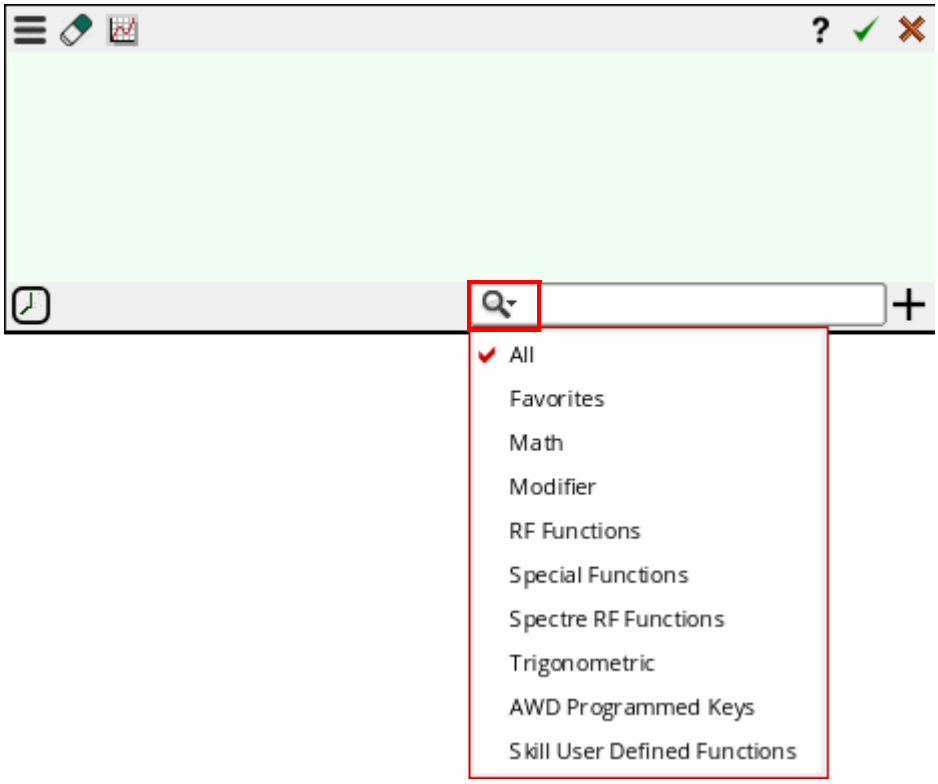


© 2023 Cadence Design Systems, Inc.

### Using the Existing Function Templates

To open the complete list of functions available in the Calculator within different function categories,

- In the *Locate Function* search field, click the drop-down arrow. All the function categories are listed, as shown in the figure below. Now, select the function category from the drop-down list and then specify a function name you want to apply. The template for the selected function is displayed.

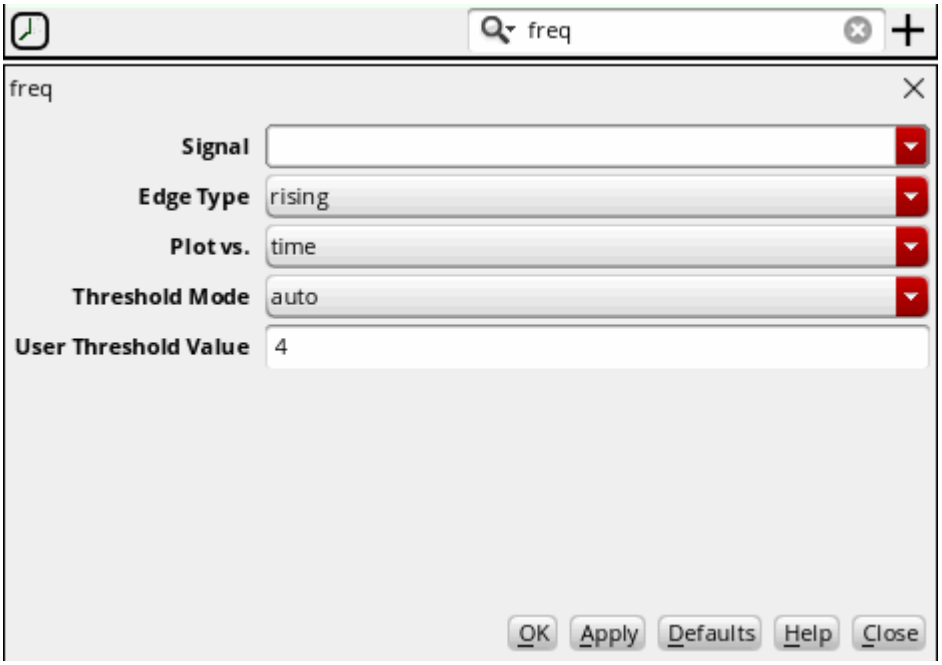



Alternatively, to view the template of a specific function,


© 2023 Cadence Design Systems, Inc.

1. Type the function name in the Locate function field. All the functions matching the specified function name are listed in the drop-down list.

2. Click the function you want to choose. For example, if you select `freq`, the template for the `freq` function appears, as shown in the figure below:



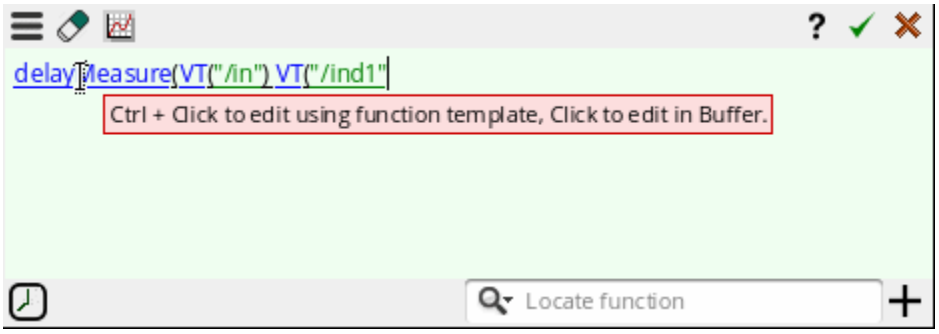

3. Specify the argument values and click *OK*. The signal you want to specify as an input should be available in the buffer of the Expression Builder.

   The specified function is added to the current cursor position in the Buffer.

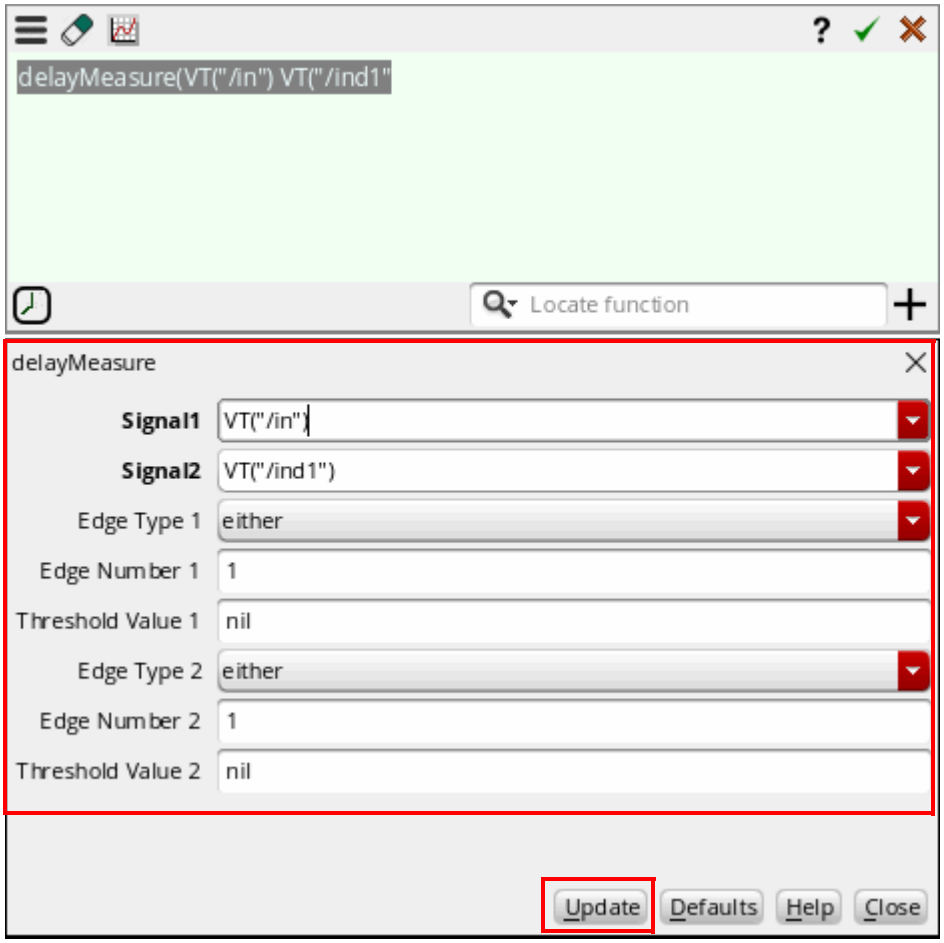
Alternatively, you can open the template for a function by pressing the `Ctrl` key and simultaneously clicking the function name in the Buffer. For example, see the figure below, if


© 2023 Cadence Design Systems, Inc.

you press the `Ctrl` key and move the pointer over the function name in the Buffer, the function expression becomes a hyperlink.




Then, click the function name to open its template with the argument values you specified in the expression. The arguments that are not specified in the expression are displayed with their default values.



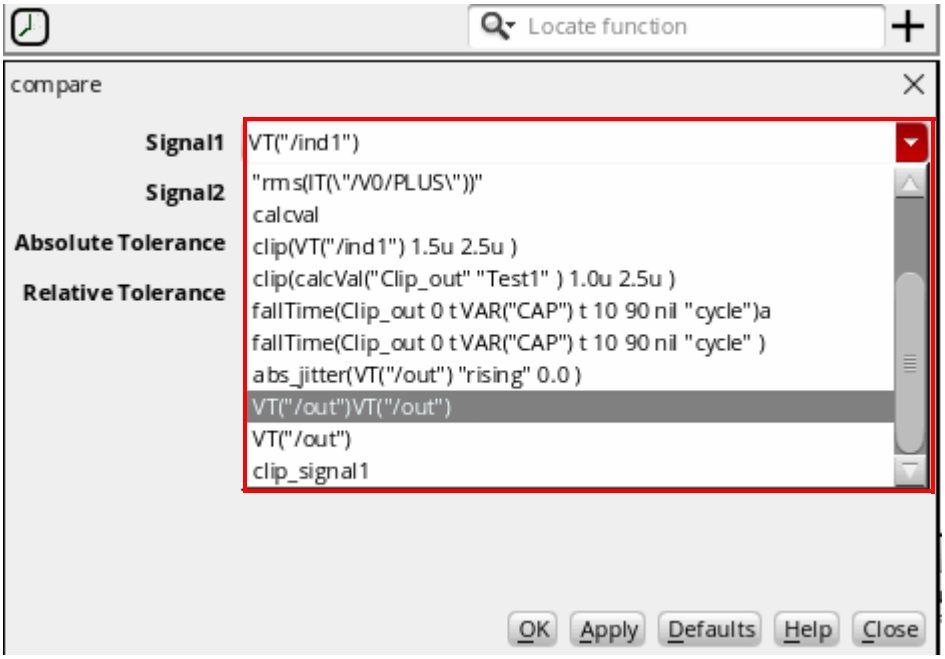


© 2023 Cadence Design Systems, Inc.

You can then update the argument values in the template and click the *Update* button. The expression is updated in the Buffer.

### Using Function Templates to Specify Input Signals

While building expressions in the buffer, if a function requires more than one input signal, for example, `compare`, all the signals that are available in the design are listed in the Auto-complete drop-down list. However, if you use function templates to build the expression, ensure that those signals are available in the Buffer and in the *Recent* drop-down list of the Expression Builder. The *Recent* drop-down list maintains a history of all the previous signals and expressions that you created using the Expression Builder. All the signals available in this drop-down are also listed in the input signal field drop-down list of the function template. Therefore, in this case, you can select the first input signal using the Buffer and the next input signal using the available history, as shown in the figure below.

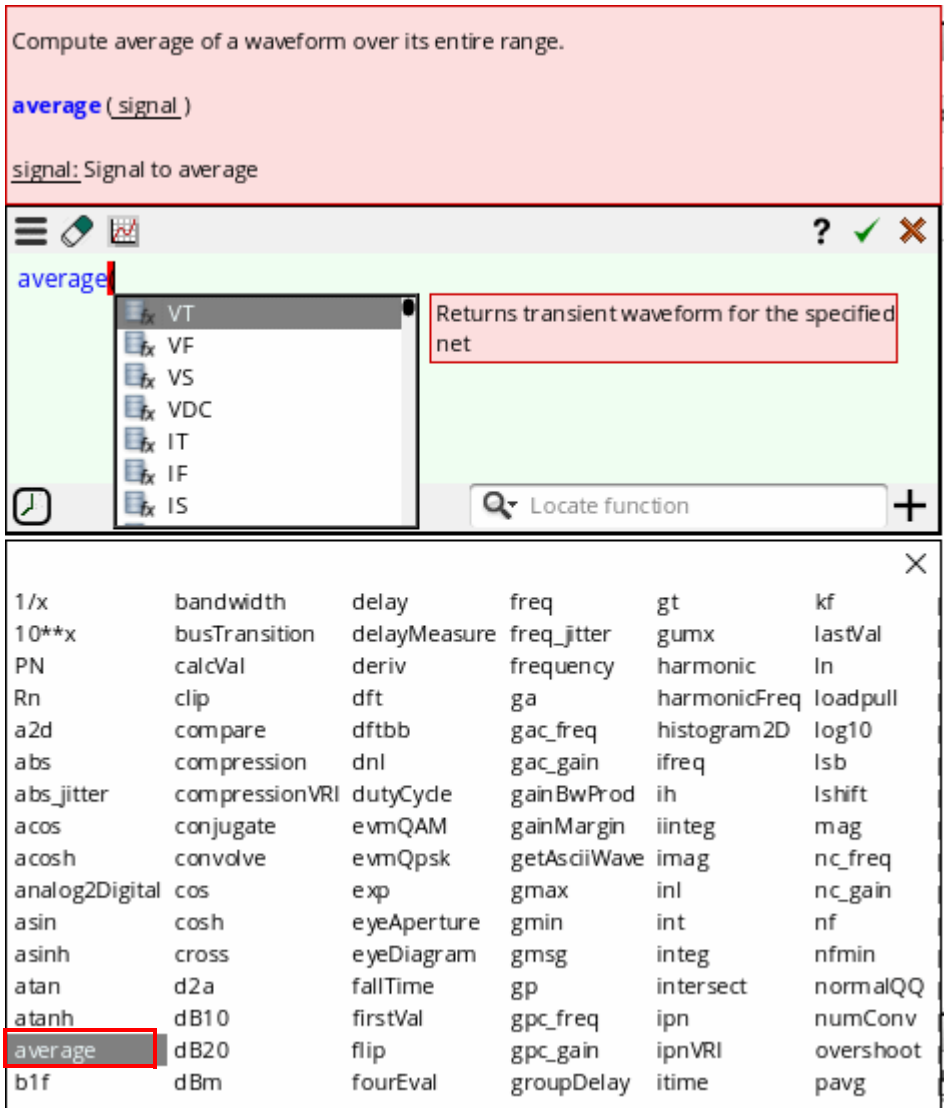




© 2023 Cadence Design Systems, Inc.

### Using Functions with a Signal Argument

If a function includes only a signal argument, the function is added at the current cursor position in the Buffer. For example, if you select *average*, the function is added to the buffer, as shown in the figure below.

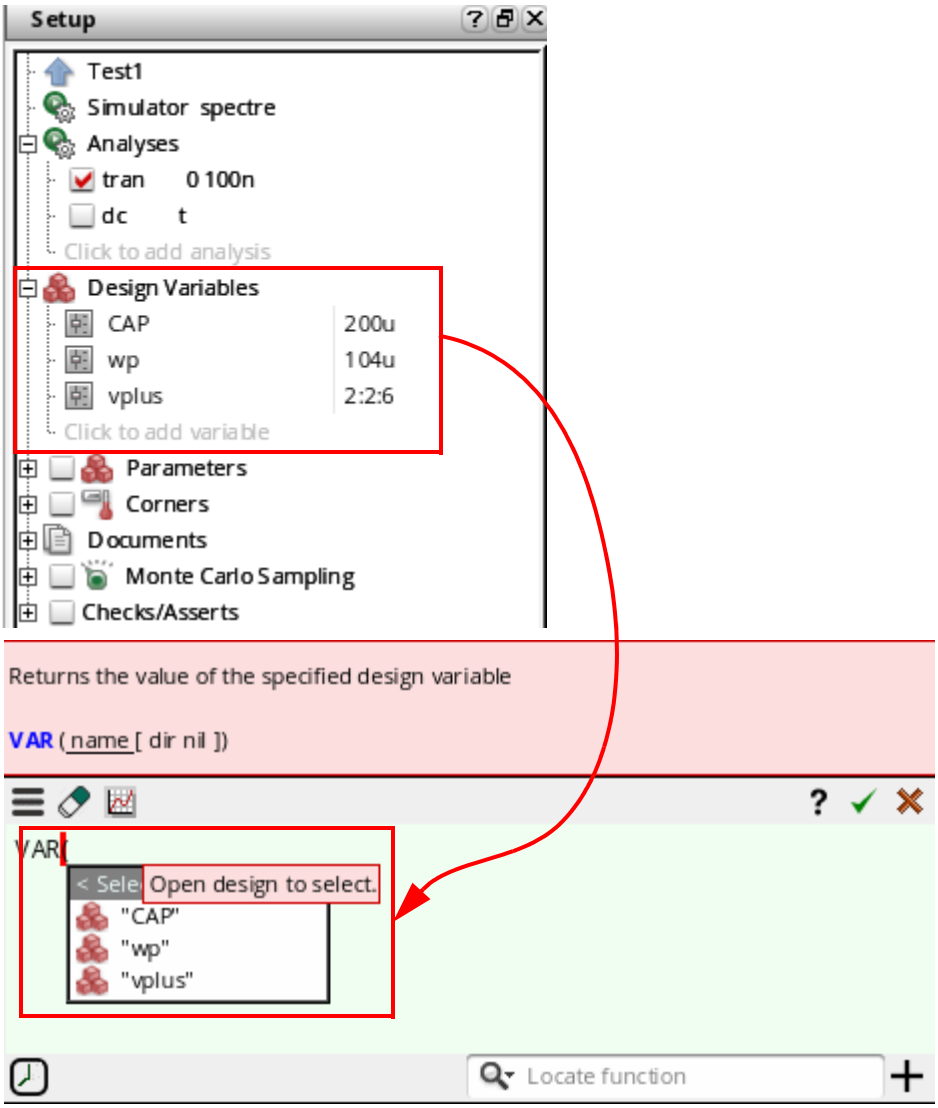



Then, you can use the Auto-complete drop-down list to build the rest of the expression.


© 2023 Cadence Design Systems, Inc.

### Using VAR in Expressions

You can also use the `VAR` syntax to specify the existing design variables while creating expressions in the Expression Builder. When you type `VAR`, the available design variables are listed in the Auto-complete drop down list.

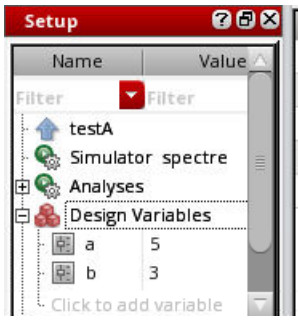



### *Using Conditional Statements*

You can use an if-else statement in output expressions to assign different values to a variable depending on the condition being satisfied.


© 2023 Cadence Design Systems, Inc.

In the following setup, `testA` has two design variables, `a` = `5` and `b` = `3`. Variables `A` and `B` are expressed using VAR syntax. Variables `C1`, `C2`, and `C3` are expressed using conditional statements. The values of variables `A` and `B` are `5` and `3`, respectively.




| Name | Type | Details | Value |
|---|---|---|---|
| A | expr | VAR("a") | 5 |
| B | expr | VAR("b") | 3 |
| C1 | expr | if((A > B) 10 20) | 10 |
| C2 | expr | if((A < B) 10 20) | 20 |
| C3 | expr | if((A != B) 30 40) | 30 |

According to the conditions defined in the expressions for variable `C1`, `C2`, and `C3`:

- If the value of variable `A` is greater than the value of variable `B`, the value of variable `C1` will be `10`, else the value will be `20`. Therefore, the value of variable `C1` = `10` (because `5` is greater than `3`).
- If the value of variable `A` is less than the value of variable `B`, the value of variable `C2` will be `10`, else the value will be `20`. Therefore, the value of variable `C2` = `20` (because `5` is not less than `3`).
- If the value of variable `A` is not equal to the value of variable `B`, the value of variable `C3` will be `30`, else the value will be `40`. Therefore, the value of variable `C3` = `30` (because `5` is not equal to `3`).

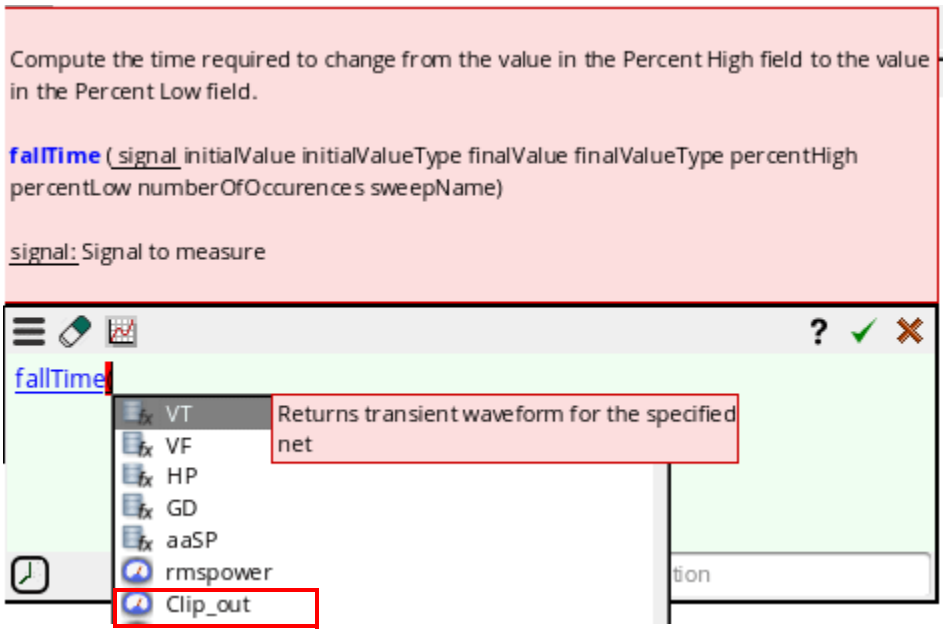
### Creating Nested Expressions

You can build expressions in the Expression Builder by using the previously-created expressions. For example, create an expression in the Expression Builder to clip the `out` signal using the `clip` function signal, and then add this expression to the buffer.

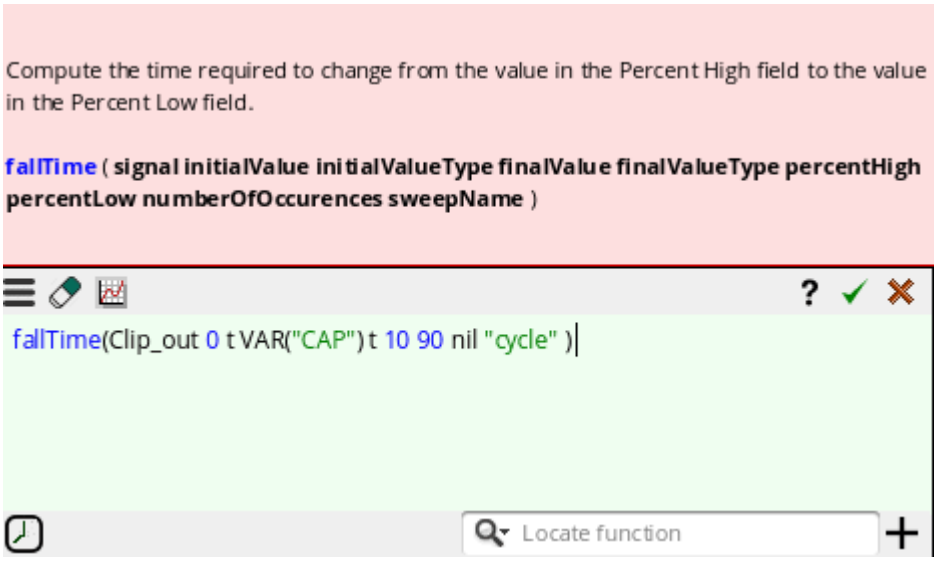
| | | |
|---|---|---|
| Clip_out | expr | clip(VT("/out") 0 4e-08) |


© 2023 Cadence Design Systems, Inc.

Now, you can specify this expression name while creating other expressions. For example, if you create an expression for `fallTime`, the `Clip_out` is also displayed in the list of available options, as shown in the figure below:




Then, build the rest of the expression, as shown in the figure below:





© 2023 Cadence Design Systems, Inc.

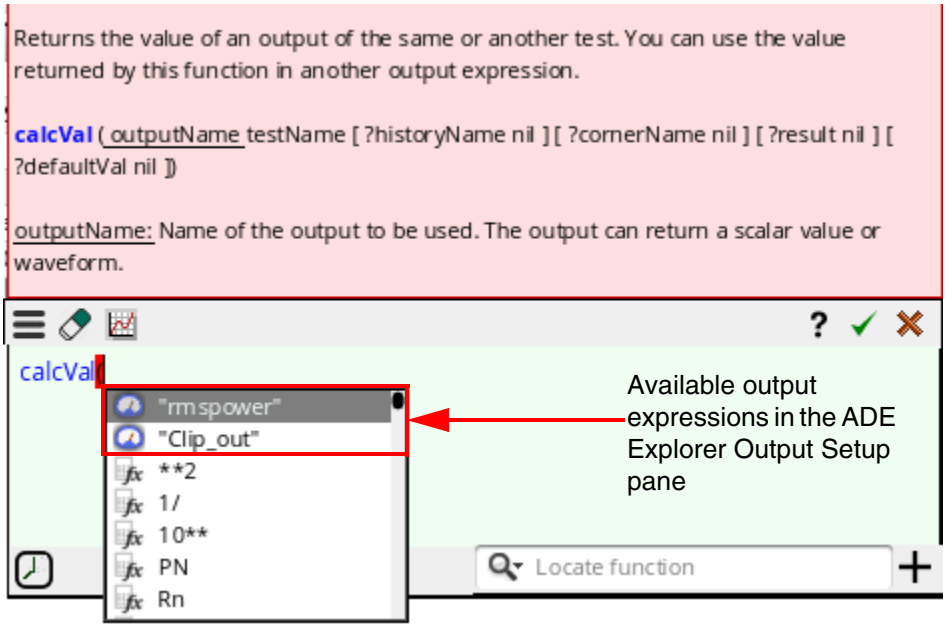
Click *OK* to send this expression to the Outputs Setup pane.

| | | | | |
|---|---|---|---|---|
| Clip_out | expr | clip(VT("/out") 0 4e-08) | ☑ | ☐ |
| | expr | fallTime(Clip_out 0 tVAR("CAP") t 10 90 nil "cycle") | ☑ | ☐ |

### Creating an Expression using CalcVal

You can type the `calcVal` function to use the result of a previously-added expression in another expression. To use `calcVal`, you need to specify the expression name, test and optionally history or corner or results name.

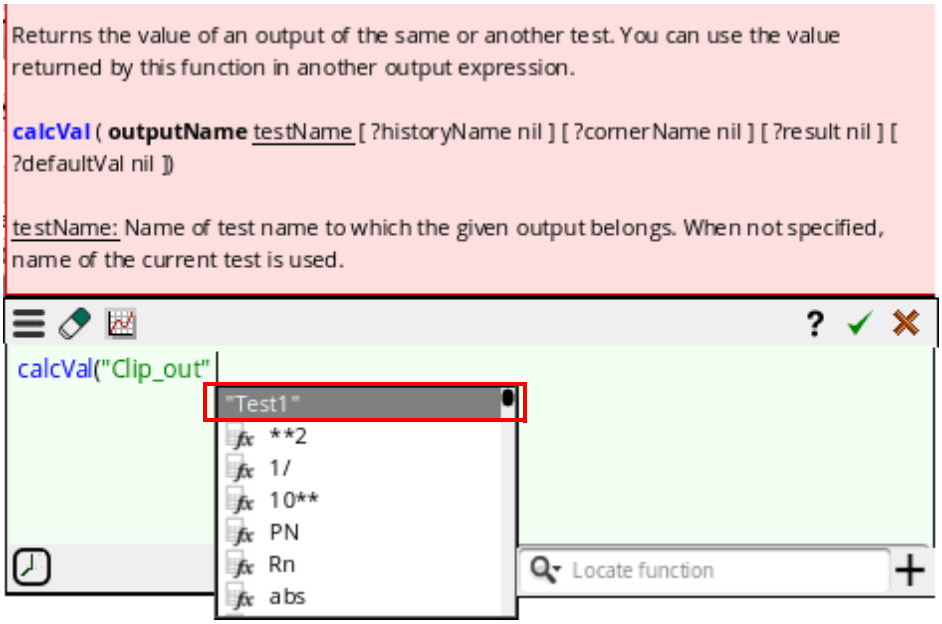
To use `calcVal`, type `calcVal` in the Expression Builder buffer. All the available output expressions are listed in the Auto-complete drop-down list. For example, in the figure below, `rmspower` and `Clip_out` output expressions are listed in the drop-down list.



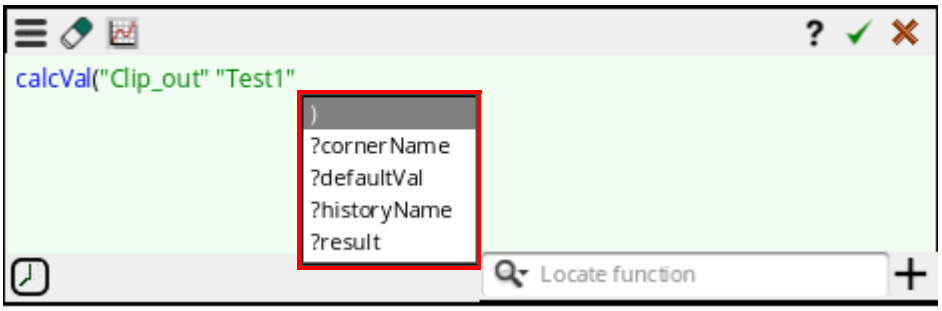


© 2023 Cadence Design Systems, Inc.

Now, after you select the output, `Clip_out`, the drop-down list displays all the test names available in the session. For ADE Explorer, only the current test name is displayed because it is a single-test environment.



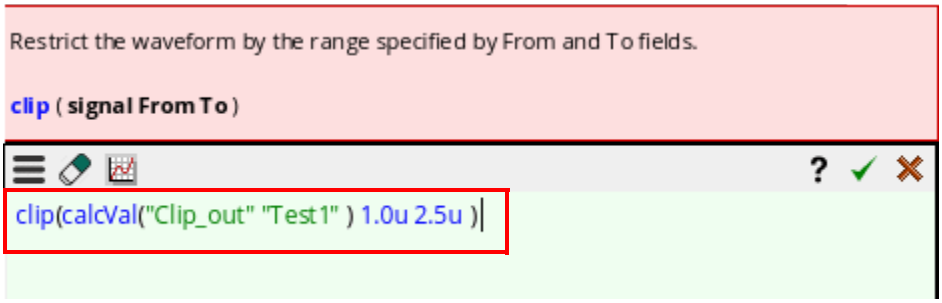

After selecting the test name, you can specify the key arguments, such as `historyName`, `cornerName`, `result` and `defaultValue`, and complete the expression. The history name must be selected before specifying the corner name. For more information about this function, see calcVal.




Upon evaluation, this `calcVal` expression returns the result of the `Clip_out` output for the specified argument values. You can use this function within other expressions.


© 2023 Cadence Design Systems, Inc.

For example, the figure below shows that the `calcVal` function is used within the `clip` function to specify the input signal. Upon evaluation, this function clips the waveform returned by the `Clip_out` according to the specified `From` and `To` values.

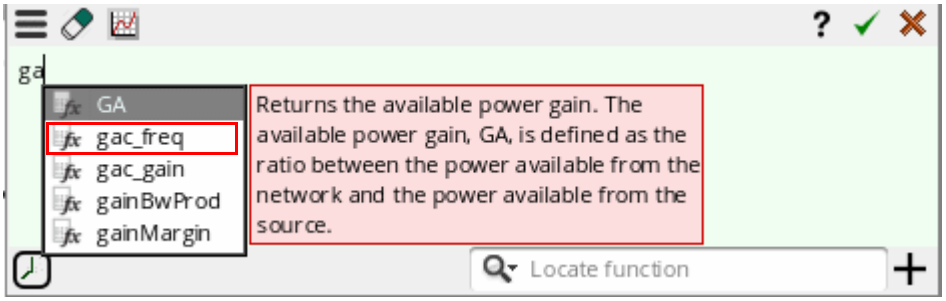


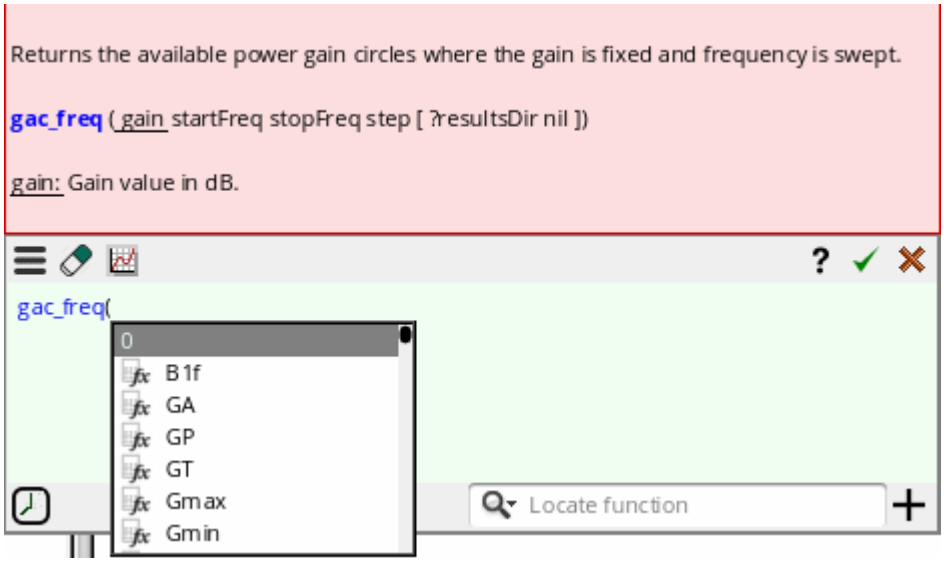

### Creating the Expression for RF Functions

You can create and evaluate expressions for RF functions in the Expression Builder. For example, consider the S-parameter (SP) analysis of the Low Noise Amplifier (LNA) design. Perform the following steps to build an expression for the `gac_freq` function using the Expression Builder:

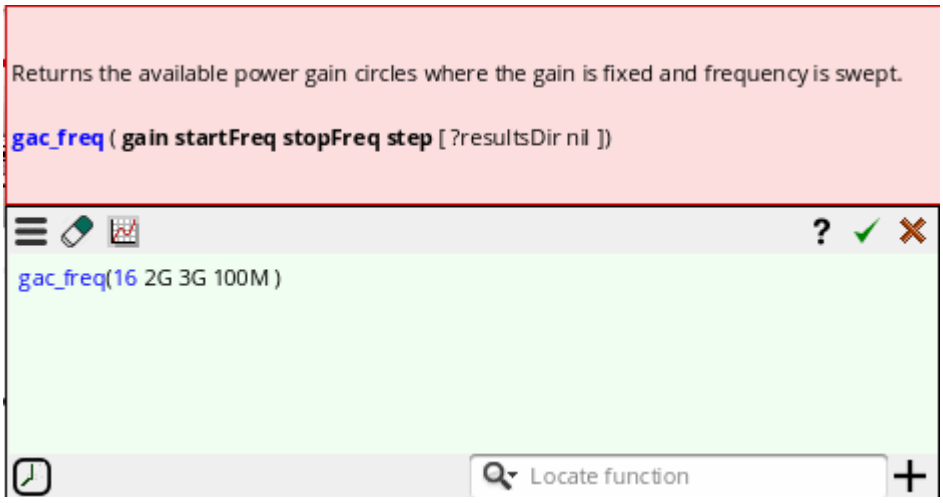

© 2023 Cadence Design Systems, Inc.

1. Select the `gac_freq` function from the Auto-complete drop-down list, as shown in the figure below:




2. Now, create an expression with `gain=16`, `startFreq=2G`, `stopFreq=3G`, and `step=100M`.

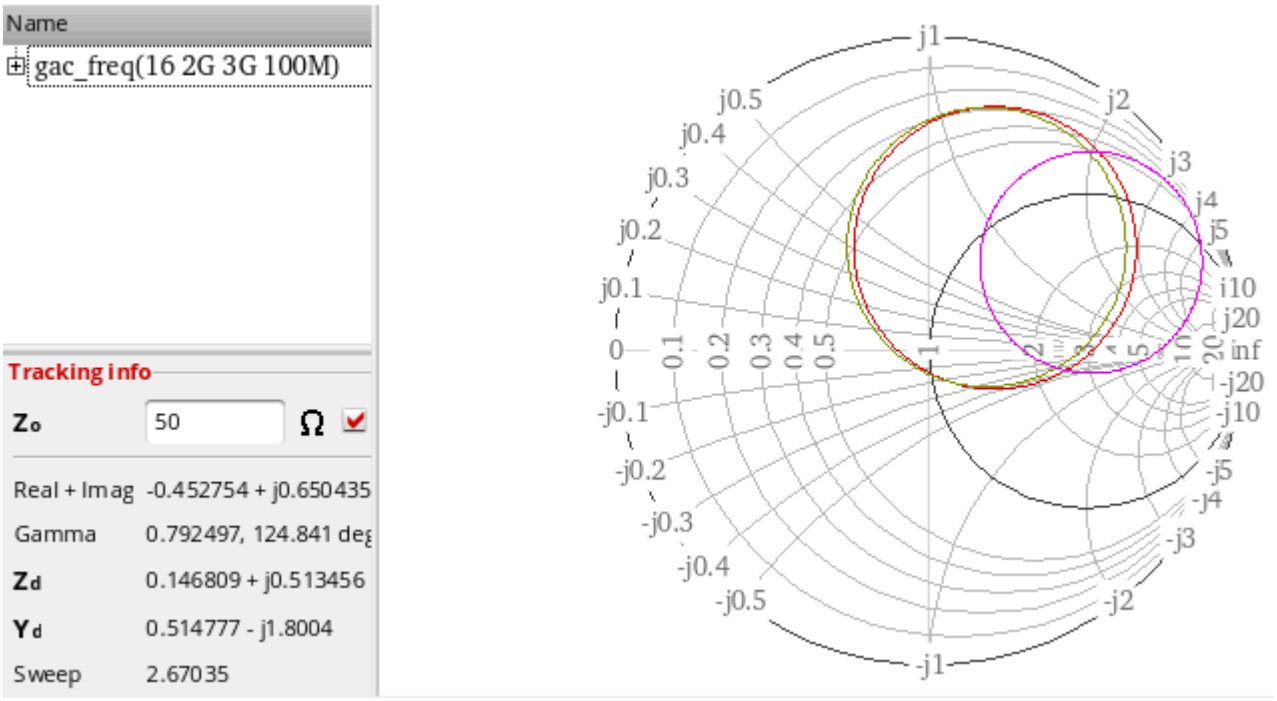




© 2023 Cadence Design Systems, Inc.

3. Click the *Evaluate* button to open the evaluation and plotting options. Select the plotting mode as `New Window` and graph type as `Impedance`, as shown in the figure below.

4. Now, click the plot icon to plot the output in a new graph window. The following plot is created:




Similarly, you can create and plot expressions for other RF functions.

### Bindkeys in Expression Builder

You can use the following bindkeys while building expressions in the Expression Builder:

- `Enter` Key—If the expression in the buffer is complete, you can press the `Enter` key to send the expression to the Outputs Setup pane. If the Expression is not complete and you press `Enter`, the cursor is moved to the next line in the Buffer.
- `Escape` Key—You can press the `Escape` key to minimize the Expression Builder window.


© 2023 Cadence Design Systems, Inc.

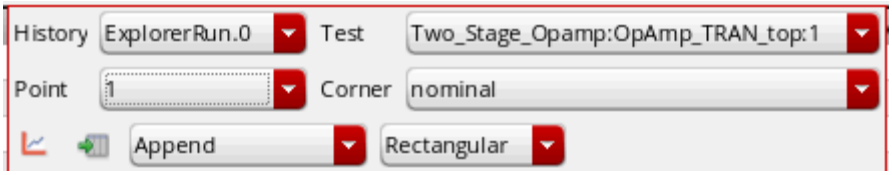
- `Ctrl`+Click—You can open a function template by holding down the `Ctrl` key and clicking the corresponding function name in the buffer.
- Up and Down Arrow Keys—If the buffer is empty, you can press the up and down arrow keys to add items saved in the *Recent* drop-down list to the buffer. If the buffer is not empty, you can press the `Shift`+ up or down arrow keys to open the *Recent* drop-down and scroll between the items. Now, if you press `Enter`, the selected item is added to the buffer.

### Evaluating or Plotting Expressions in Expression Builder

By default, the expressions you created using Expression Builder are plotted when you add them to the Outputs Setup tab and run the simulation.

However, you can also use the plotting options available within Expression Builder to directly plot expressions without running the simulation.

- Create a new expression or double-click an existing expression in the Expression Builder.
- Then, click the evaluate or plot button ( ) available on the menu bar. The following options appear:




  - *History* — Lists the histories available in the given setup. Use this drop-down list to select a history for which you want to evaluate or plot the given expression. The default history for an ADE Explorer setup is `ExplorerRun.0`. If you are using an ADE Assembler setup, then the recent history is selected by default.
  - *Test* — Lists the tests that are enabled in the given setup. For ADE Explorer, only the current test is displayed. Use this drop-down list to select a test that you want to use to evaluate or plot the given expression.

  **Note:** All the disabled tests are not listed.

  - *Point* — Lists the sweep points specified in the given setup. Use this drop-down list to select the number of points for which you want to evaluate or plot the given expression. To evaluate and plot the expression for all points, choose `All` from the drop-down.

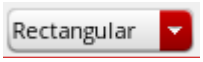

© 2023 Cadence Design Systems, Inc.

- *Corner* — Lists the corners that are enabled in the given setup. Use this drop-down list to select a corner for which you want to evaluate or plot the given expression. The default value is `nominal`. To evaluate and plot the expression across all corners, choose `All` from the drop-down.

**Note:** All the disabled corners are not listed.

- — Evaluates the given expression and plots the waveform in the graph window. If the result is scalar, it is displayed in the Expression Builder.
- — Evaluates the given expression and lists the results in Virtuoso Visualization and Analysis XL Table.
-

— Lists the plotting modes that you can select to plot waveforms. The available options are:
  - `Append`
  - `Replace`
  - `New Subwindow`
  - `New Window`

  **Note:** When you click the table button to list the results in a table, the New Subwindow and New Window plotting modes work in the same way. The outputs are listed in a new tab in the Virtuoso Visualization and Analysis Table window.
- 

— Lists the graph type that you want to use to plot the output waveform. The available options are:
  - `Rectangular`
  - `Polar`
  - `Impedance`
  - `Admittance`

## Related Blogs

- Virtuoso Video Diary: How Can I Plot or Evaluate with the New Expression Builder?
- Virtuosity:Expression Builder - Now Plots ALL!


© 2023 Cadence Design Systems, Inc.

### Disabling Expression Builder

Expression Builder is opened by default when you build expressions in ADE Explorer and ADE Explorer Outputs Setup pane. However, if you want to open the Calculator window, you can set the useExpressionBuilder environment variable in CIW, as shown below:

```
envSetVal("maestro.gui" "useExpressionBuilder" 'boolean nil)
```

### Limitations

Currently, the history items saved in the Recent drop-down list cannot be deleted. By default, maximum of 10 histories are saved in this list.


© 2023 Cadence Design Systems, Inc.

# Defining Specifications for an Output

You define performance specifications on the Outputs Setup 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 the *Results* tab. 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.

## Defining and Managing Specifications for Outputs

You can define a performance specification for a measurement in the *Spec* column on the Output Setup pane.

By default, the specification defined for a measurement applies to all the corners enabled for the test. You can also disable the corner specification.

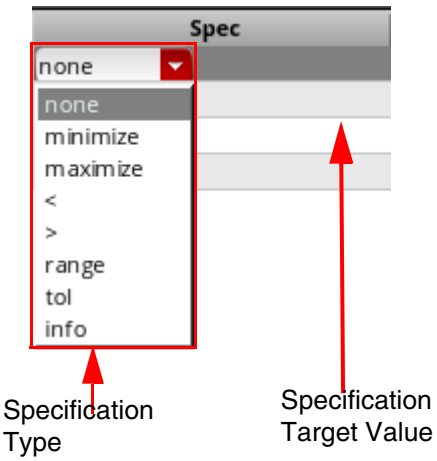
## Defining Specification for an Output

To define a specification for a measurement on the Outputs Setup tab, do the following:

1. Double-click in the *Spec* column.


© 2023 Cadence Design Systems, Inc.

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 select one of the following specification types and fill in the right half of the cell with an appropriate target information as indicated:

**Table 4-1 Specification Types**

| Objective | Target | Description |
|---|---|---|
| *minimize* | `targetValue` | Minimize the result; used in circuit optimization |
| *maximize* | `targetValue` | Maximize the result; used in circuit optimization |
| *<* | `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 |

**Note:** A target specification value can be specified in the form of constant values or expressions using global variables or other output names. For more details, refer to Using Variables in Specifications.


© 2023 Cadence Design Systems, Inc.

3. (Optional) In the *Weight* column next to the specification, specify the weighting factor for the specification.

### Using Variables in Specifications

The target values for specifications can be specified as a constant or an expression using variables. In case the value is specified as an expression, the evaluated value of the expression is compared with the output value to identify the *pass* or *fail* status. By using expressions in specifications, you can vary the specifications across sweep points.

The expressions can be created for specification target values in any of the following ways:

- **By creating calculator expressions**: An example is given below.

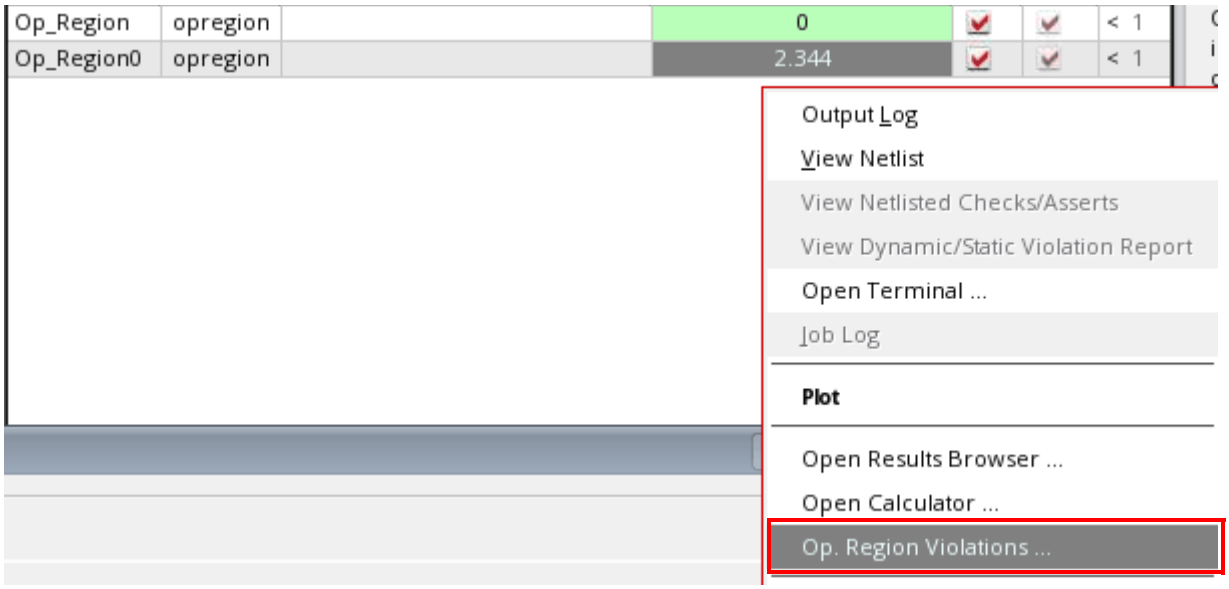
  The field for specification value is a multiline text field that is expanded automatically to accommodate the complete expression.

- **By using a SKILL or OCEAN script**: For this, you can add the SKILL or OCEAN code in the *Spec* column. Alternatively, you can create a separate output with a SKILL or OCEAN script. Result of that output can be referred in the expression for a specification type.


© 2023 Cadence Design Systems, Inc.

*Important*

If you run Op Region expressions in Assembler and then switch to Explorer, an additional option, *Op. Region Violations*, appear in the context-menu when you right-click in the Value column of the opregion output row (as shown in the figure below).




When you select this option, the *Operating Region Specification* form appears. For more information about this form, see *Virtuoso ADE Assembler User Guide*.

## Saving Violation Filter

See Performing Circuit and Device Checks

## Saving Operating Point Parameters

In the Outputs Setup pane, you can specify the instances and their operating point parameters that you want the simulator to save. Currently, this feature is available only in the Spectre and operating point parameters for only DC and transient analyses are saved.

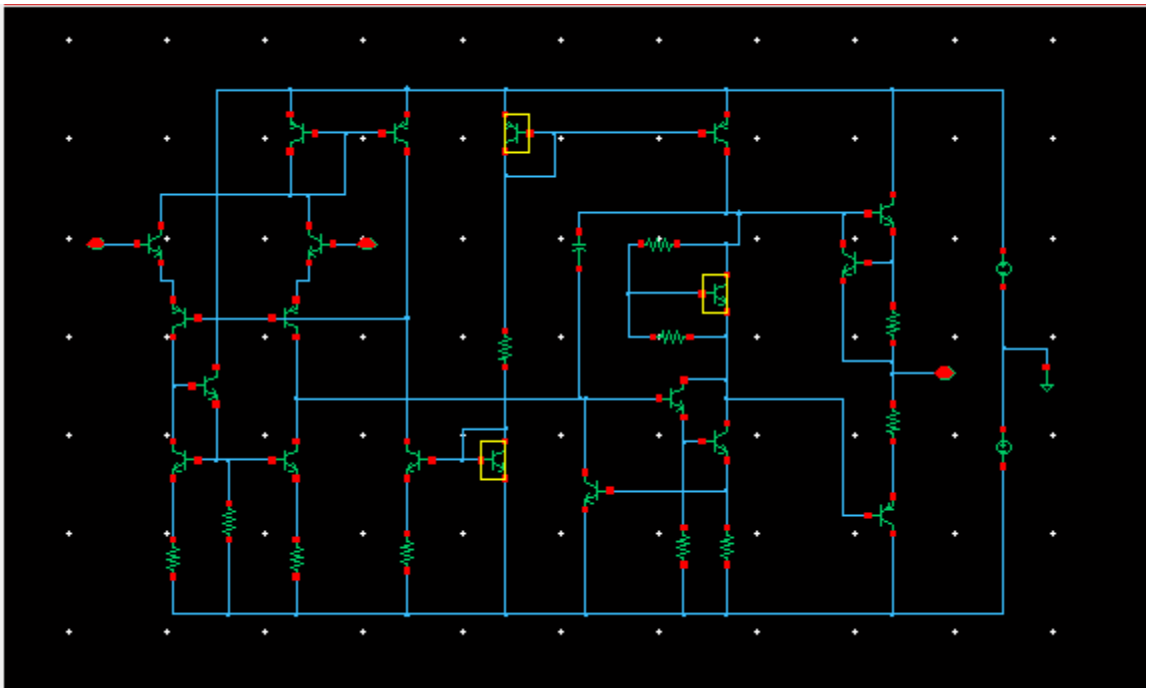
**Note:** This output is supported only for the Spectre and AMS simulators.

By default, the instance column on the Outputs setup pane is blank. To add instances and their operating point parameters that are to be saved, perform one of the following steps:

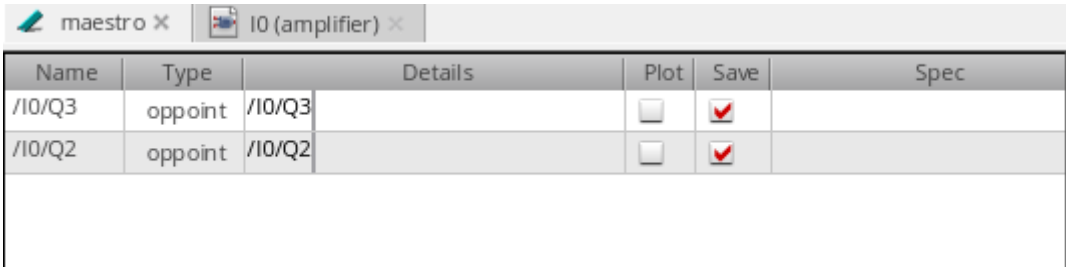
- Choose *Outputs – To Be Saved – Select By Op Parameters*.


© 2023 Cadence Design Systems, Inc.

The Schematic Editor window appears. Select the required instances in this window. The selected instances are highlighted in the Schematic Editor window, as shown in the figure below.



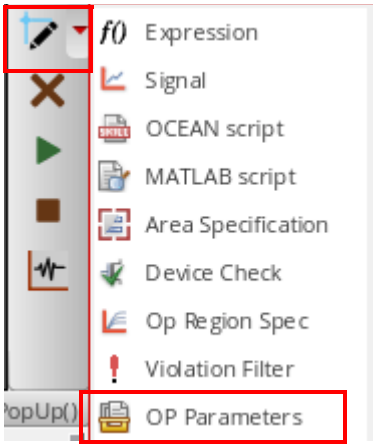
The selected instances are listed in the table as shown in the figure below:



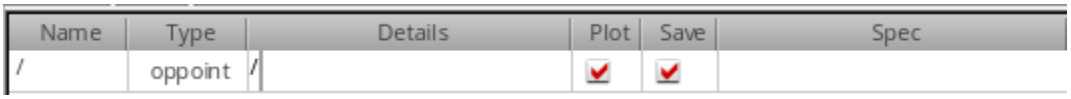


© 2023 Cadence Design Systems, Inc.

- On the Run toolbar, click the *Add Outputs* button and choose *OP Parameters*, as shown in the figure below.



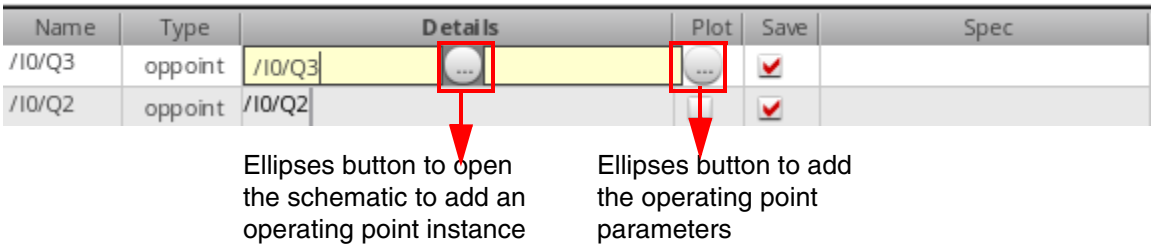

An empty row is added in the Outputs Setup pane as shown in the figure below:




### Adding Operating Point Parameters

In the *Details* column, do one of the following to specify the operating point parameters to be saved for each instance:

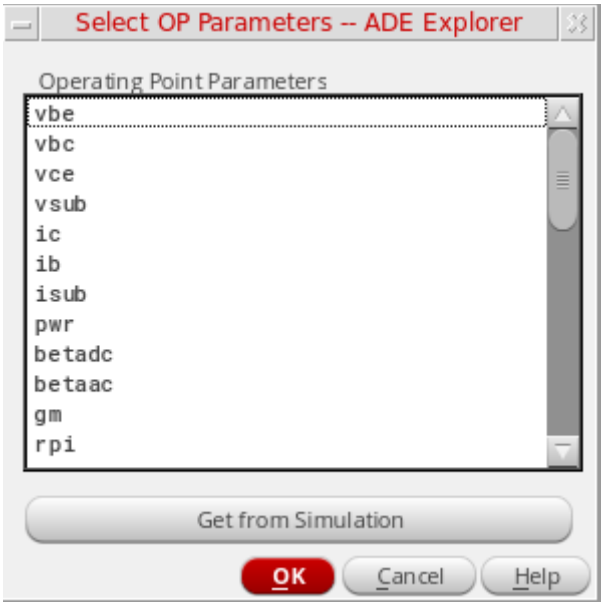
- Double-click in the *Details* column for each instance and type the name of the operating point parameters that are to be saved. The specified parameters must exist for that instance.
- Click the ellipses (...) button adjacent to the cell in *Details* column, as shown in the figure below.





© 2023 Cadence Design Systems, Inc.

The Select OP Parameters form appears. Choose the operating point parameters from the Select OP Parameters form. If the instance is not selected or invalid, the Save OP Parameters form will not open and a warning message is displayed in CIW.




If you do not find the operating point parameters listed in the form, click the *Get from Simulation* button to run the simulation. If the simulation run is successful, the operating point parameters are listed in the form; otherwise, an error message is displayed in CIW.

Next, in the Select OP Parameters form, select the operating point parameters that you want to save by using the `Ctrl` key and click *OK*. The selected operating point parameters are added to the table.

| Name | Type | Details | | Value | Plot | Save | Spec |
|---|---|---|---|---|---|---|---|
| /I0/Q3 | oppoint | /I0/Q3 | ib gm rpi | | ☐ | ☑ | |
| /I0/Q2 | oppoint | /I0/Q2 | ib | | ☐ | ☑ | |

**Note:** Consider a scenario in which you have added two instances, `/I0/Q3` and `/10/Q2`, to the table from the schematic. You open the Select OP Parameters form for `/I0/Q3` and do not find anything listed in this form. In such a scenario, you need to click the *Get from Simulation* button to retrieve operating point parameters from the simulation run. If you now open the Select OP Parameters form for instance `/I0/Q2`, you will find that the operating point parameters for this instance are already listed in the form and you do not need to click the *Get from Simulation* button again.

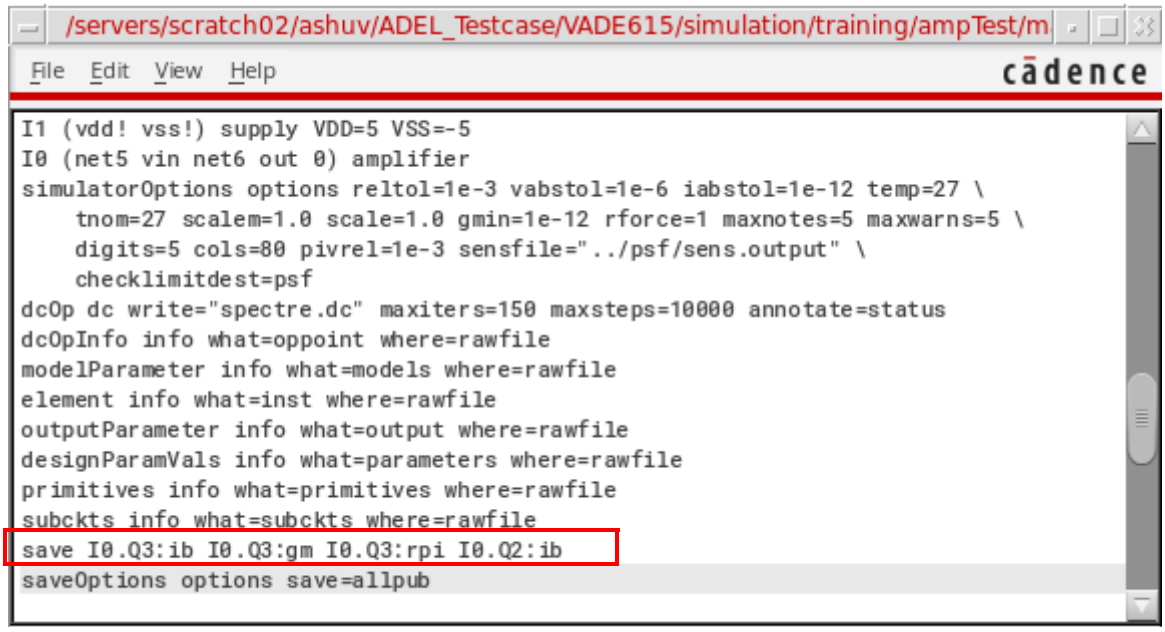
If you next add some more instances from schematic to the table and open the Select Operating Points form for one of them, the operating point parameters for these


© 2023 Cadence Design Systems, Inc.

instances will not be listed in the form and you will need to click the *Get from Simulation* button again to retrieve operating point parameters for the newly added instances.

The operating point parameters that have already been added in the *Operating Parameters* field are always selected in the Select OP Parameters form if present.

Run the simulation. To ensure that the operating point parameters were saved, open the `input.scs` file (the link to this file is provided in the netlist). For example, in the figure below, you can see that the `ib`, `gm`, and `rpi` operating point parameters were saved for instance `M3`, and `ib` operating point parameter was saved for instance `Q2` in the `input.scs` file.



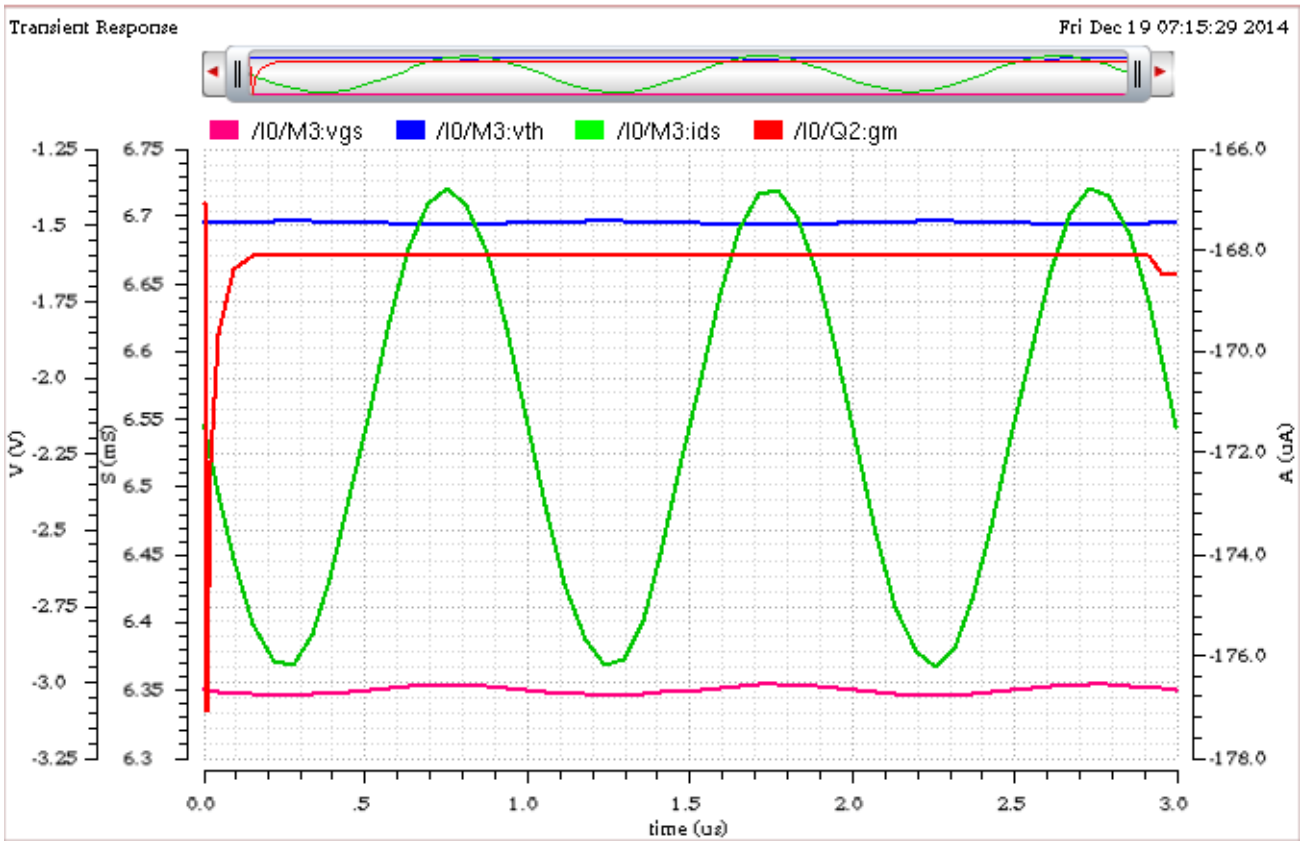

**Note:** If an entry corresponding to an operating point is found in the `opParamExprList` or `optParamExprList` CDF parameter, the operating point appears as both an inline subcircuit and a regular subcircuit in the `save` statement in the `input.scs` file, as shown below:

```
save I0.Q3:ib I0.Q3:gm I0.Q3.bjt4anpn:gm I0.Q3:rpi I0.Q2:ib
saveOptions options save=allpub
```

If you are using the Spectre simulator, you may see a warning message in case the inline subcircuit syntax does not exist in the design, but you can safely ignore that warning.


© 2023 Cadence Design Systems, Inc.

You can click the *Plot* button to plot the operating point parameters for instances `Q2` and `M3`. The selected parameters for these instances are plotted in append mode.



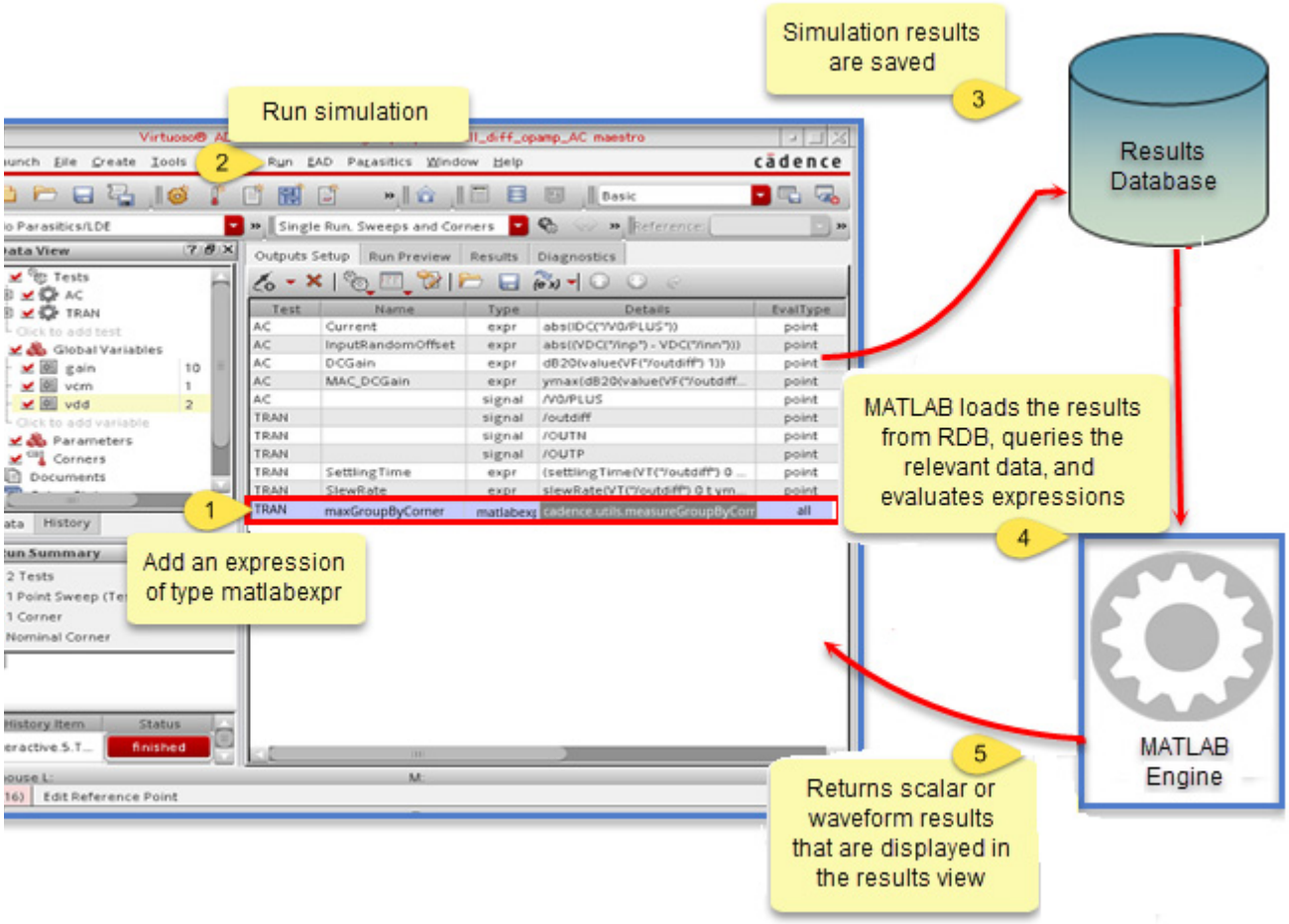

**Note:** You can also add and delete operating points by using the saveOpPoint and deleteOpPoint OCEAN commands.


© 2023 Cadence Design Systems, Inc.

# Using MATLAB® Expressions in Outputs

Fast, accurate, and efficient analysis of simulation results is one of the key requirements for circuit simulation and verification. Integration of MATLAB® with ADE Explorer provides the capability to apply various data slicing and querying methods for advanced data analysis and waveform processing using a high-level language.

To use MATLAB in ADE Explorer, you can add MATLAB expressions as outputs that are evaluated in the MATLAB engine. The MATLAB engine runs independent of ADE Explorer, but there is a channel through which ADE Explorer exposes the circuit simulation data. This provides MATLAB a capability to access both the scalar and waveform data that can be used in measurements for single points, sweeps or corners as well as for Monte Carlo analysis. The results of these expressions are returned by MATLAB and displayed on the *Results* tab in ADE Explorer.





© 2023 Cadence Design Systems, Inc.

**Note:** To evaluate MATLAB expressions, you need to run MATLAB vR2018a or later. You also require the Virtuoso MATLAB Integration Option (95252) license.

For more information about how you can use MATLAB expressions, see Using MATLAB Expressions in Outputs in *Virtuoso ADE Assembler User Guide*.

# Deleting Outputs

To delete an output, do one of the following in the Output Setup tab:

- Select the output row and choose *Outputs – Delete*.
- Select the output row and click the button on the Run toolbar.
- Right-click the output row and choose *Delete Output*.

  The selected output row is deleted.

## Adding a User-Defined Column

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 add a user-defined column in the *Outputs Setup* tab:

Right-click the title of any existing column and choose *Add Column*.

The Add Column form appears.

1. In the *New Column Name* field, specify a name for the new column to be added.
2. Click *OK*.

   A new column is added to the right of the existing columns.

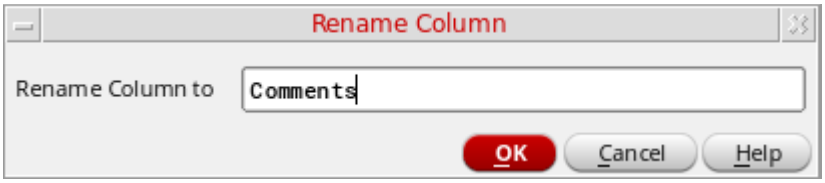

© 2023 Cadence Design Systems, Inc.

## Renaming a User-Defined Column

To rename a user-defined column, do the following:

1. Right-click the column in the *Outputs Setup* tab and choose *Rename Column*.

   The Rename Column form appears.




2. In the *Rename Column to* field, specify a new name for the column.
3. Click *OK*.

## Deleting a User-Defined Column

To delete a user-defined column, Right-click the user-defined column and choose *Delete Column*.

The column is removed from the *Outputs Setup* tab.

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.


© 2023 Cadence Design Systems, Inc.

## Filtering and Searching in Outputs Setup Pane

You can use the search filters (*Filters*) displayed at the top of the Outputs Setup pane to filter the outputs based on specified search words or strings. These filters are displayed for all the fields in the Outputs Setup pane except for the *Plot* and *Save* check boxes.

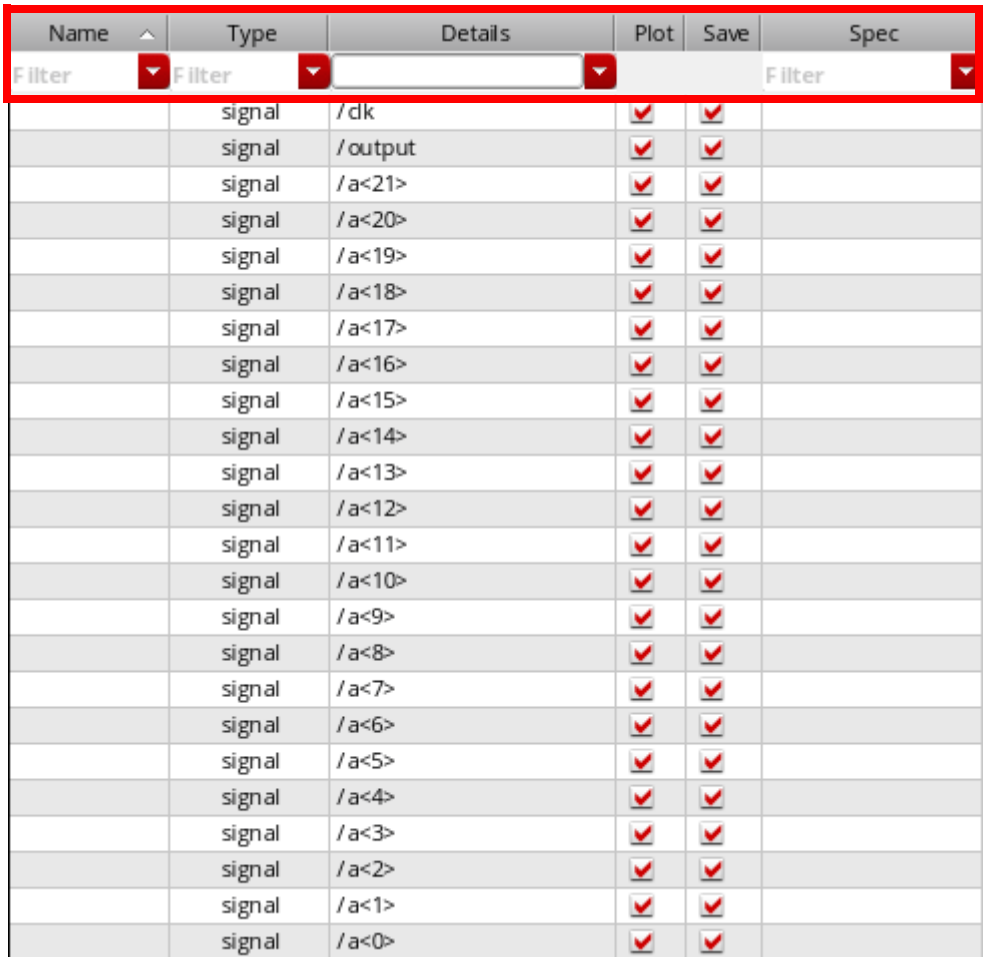
**Note:** By default, filters are displayed only when `20` or more output rows are specified in a given ADE Explorer setup. However, you can increase or decrease this limit (threshold) by setting the outputSetupFilteringNumOutputThreshold environment variable.

Regardless of the setting of this variable, you can also show or hide filters manually for any number of output rows. To do this, right-click anywhere on the outputs header and choose *Show Filter*s or *Hide Filters*.


© 2023 Cadence Design Systems, Inc.

Video

For video demonstration, see Filtering Data on the Outputs Setup and Results Pane on Cadence Online Support.



| Name | Type | Details | Plot | Save | Spec |
|---|---|---|---|---|---|
| Filter | Filter | | | | Filter |
| | signal | /clk | ✓ | ✓ | |
| | signal | /output | ✓ | ✓ | |
| | signal | /a<21> | ✓ | ✓ | |
| | signal | /a<20> | ✓ | ✓ | |
| | signal | /a<19> | ✓ | ✓ | |
| | signal | /a<18> | ✓ | ✓ | |
| | signal | /a<17> | ✓ | ✓ | |
| | signal | /a<16> | ✓ | ✓ | |
| | signal | /a<15> | ✓ | ✓ | |
| | signal | /a<14> | ✓ | ✓ | |
| | signal | /a<13> | ✓ | ✓ | |
| | signal | /a<12> | ✓ | ✓ | |
| | signal | /a<11> | ✓ | ✓ | |
| | signal | /a<10> | ✓ | ✓ | |
| | signal | /a<9> | ✓ | ✓ | |
| | signal | /a<8> | ✓ | ✓ | |
| | signal | /a<7> | ✓ | ✓ | |
| | signal | /a<6> | ✓ | ✓ | |
| | signal | /a<5> | ✓ | ✓ | |
| | signal | /a<4> | ✓ | ✓ | |
| | signal | /a<3> | ✓ | ✓ | |
| | signal | /a<2> | ✓ | ✓ | |
| | signal | /a<1> | ✓ | ✓ | |
| | signal | /a<0> | ✓ | ✓ | |

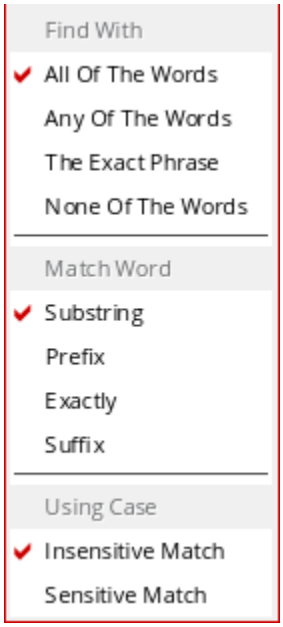
To filter the output rows in the Outputs Setup pane, do one of the following:

- Type search words or strings in the *Filter* drop-down.
- Select the previously searched words or strings that are saved in the *Filter* drop-down. For more information, see Viewing Search History.
- Select the saved named filter from the *Named filters* toolbar. For more information, see Using Named Filters Toolbar.

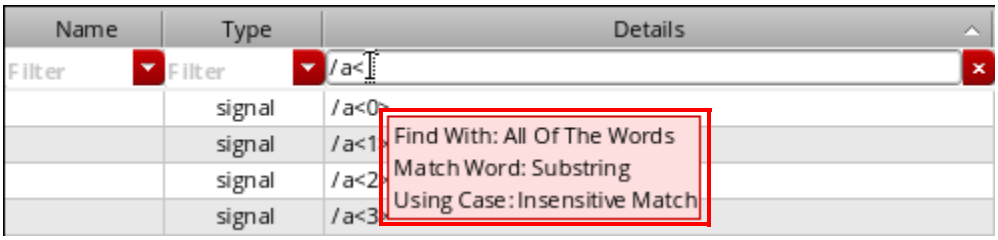
To specify filter conditions while searching or filtering the outputs,


© 2023 Cadence Design Systems, Inc.

➡ Right-click in the *Filter* drop-down list and select the conditions listed in the figure below. This figure displays the conditions that are selected by default.




The tooltip on the *Filter* drop-down list displays the selected search conditions, as shown in the figure below.

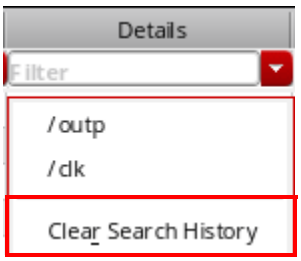



### Viewing Search History

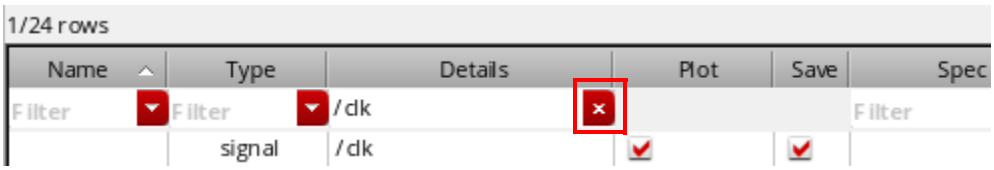
A history of the searched words or strings is also maintained in the *Filter* drop-down list. This history can be reused later, and therefore, helps in making the filtering or searching faster for the frequently used outputs. In addition, you do not need to type the search text again to search for particular outputs.


© 2023 Cadence Design Systems, Inc.

To clear the search history, select *Clear Search History* from the drop-down list of search filter.




In addition, after the searched outputs are filtered and listed, the *Filters* drop-down list changes into a Cancel button that you can use to cancel the search and display all the specified outputs.




### Using Named Filters Toolbar

You can use the *Named filters* toolbar to create and save named filters for combination of column searches that you use or refer to more frequently. These named filters include search words or strings used to filter the outputs for all columns.

To open the *Named filters* toolbar,

- Right-click anywhere on the menu bar and choose *Toolbars – Named filters*.
- Right-click anywhere on the toolbar area and choose *Named filters*.

To save a named filter,

➡ Type a name in the drop-down list and click the *Save named filter* button.

The specified name is added as a named filter and displayed in the drop-down list.

To delete a saved filter,


© 2023 Cadence Design Systems, Inc.

➡ In the drop-down list, select a named filter that you want to delete and click the *Delete named filter* button.

The selected named filter is deleted.


© 2023 Cadence Design Systems, Inc.

# Importing and Exporting Outputs to a CSV File

Working on a large number of signals and expressions, which includes specifying values and setting plot and save options for signals and expressions, in the *Outputs Setup* tab in ADE Explorer is difficult. Therefore, to manage the signals and expressions better, you can export the outputs from the *Outputs Setup* tab to a CSV file. You can then edit or add output values to the file in a text editor and import the file back to ADE Explorer.

This section covers the following topics:

- Structure of a CSV File on page 467
- Exporting Outputs on page 468
- Importing Outputs on page 469

## Structure of a CSV File

The following table lists the output components in the order required by the CSV file format and provides a brief description of each component.

| Component | Description |
|---|---|
| *Name* | Indicates the name of the expression.<br>**Note:** If the output is for a signal, specify `nil` for this component. |
| *Type* | Indicates the type of output. The possible values are:<br>■ `net` — Indicates a node voltage<br>■ `terminal` — indicates a current<br>■ `expr` — indicates an expression |
| *Output* | Indicates the name of the net, terminal or the expression. |
| *Plot* | Indicates if a output is to be plotted or not. The possible values are:<br>■ `t` — Indicates output is to be plotted<br>■ `nil` — Indicates output is not to be plotted |


© 2023 Cadence Design Systems, Inc.

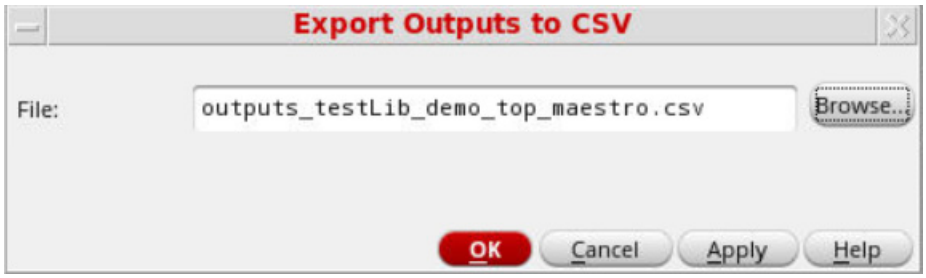
| | |
|---|---|
| *Save* | Indicates whether an output is to be saved or not. The possible values are:<br>■ `t` — Indicates that the output is to be saved<br>■ `nil` — Indicates that the output is not to be saved |

## Exporting Outputs

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. Choose *Outputs – Export*.

   The *Export Outputs to CSV* form appears.

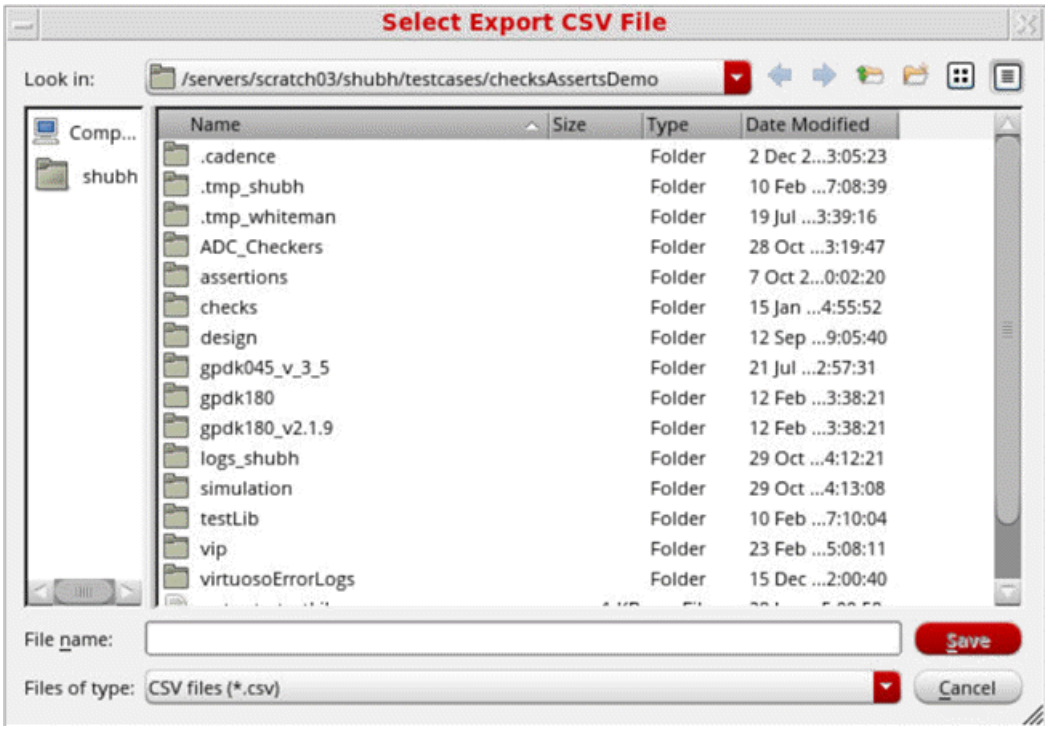



3. Specify a name for the CSV file to which the output is to be saved.

   The default file name is `outputs_<lib>_<cell>_<view>.csv` where `lib`, `cell`, and `view` are the library name, cell name, and view name associated with the test. By default, it is saved in the testcase directory.


© 2023 Cadence Design Systems, Inc.

Alternatively, click *Browse* and choose a directory and name for the CSV file.




4. Click *OK* to export the outputs.

All the outputs defined on the *Outputs Setup* tab are saved in the specified CSV file. You can open the CSV file in any text editor.

## Importing Outputs

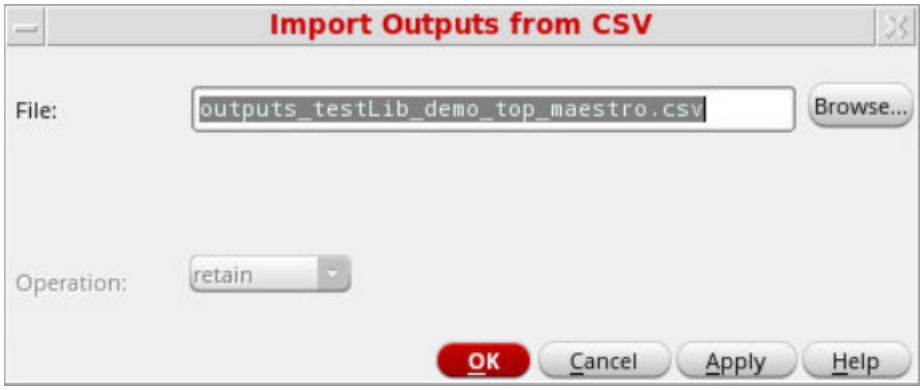
To import outputs from a CSV file to the *Outputs Setup* tab, do the following:

1. Choose *Outputs – Import*.

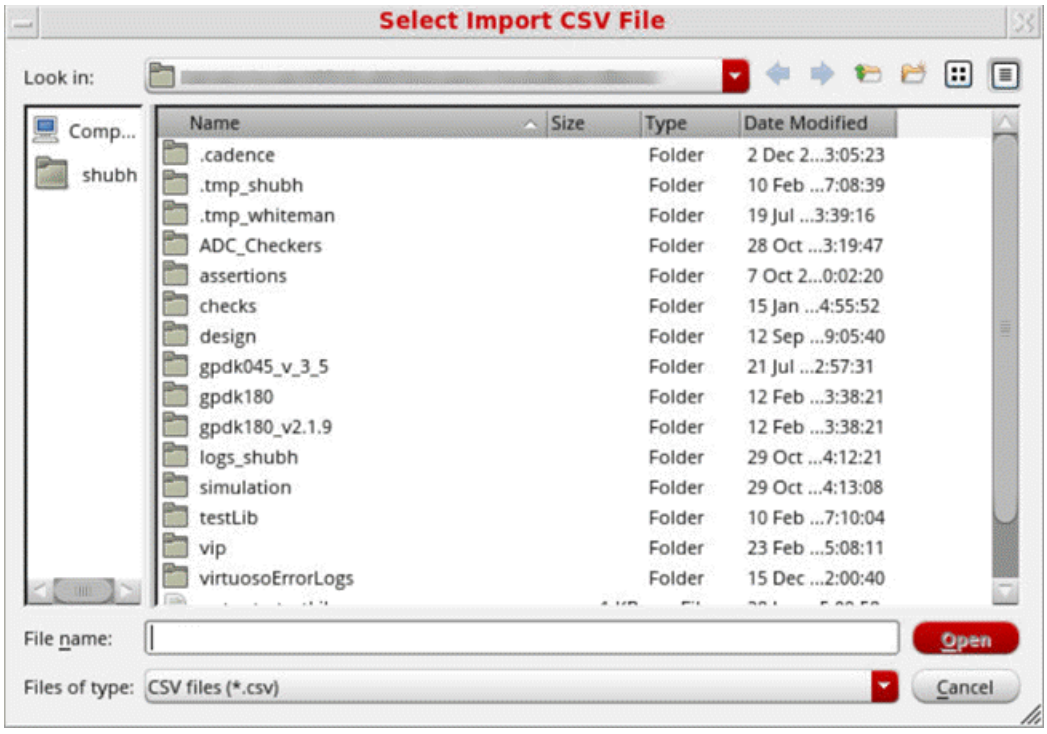
   The *Import Outputs from CSV* form appears.


© 2023 Cadence Design Systems, Inc.

The default file name appears in the *File* field.




Click *Browse* to locate and select a different file.




**Note:** Only those outputs that are not defined in the *Outputs Setup* pane are loaded from the CSV file being imported. Therefore, to ensure that the changes made to an exported output are visible in the *Outputs Setup*, delete that output before importing the CSV file.

2. Click *OK* to import the outputs.

The imported output value is displayed in the *Outputs Setup* pane of ADE Explorer.


© 2023 Cadence Design Systems, Inc.

***Important Points to Note:***

- 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 setup.
- Outputs that include the following are ignored while exporting outputs to CSV:
  - OCEAN measures
  - MATLAB scripts
  - area specifications
- By default, the Test, Name, Type, Output, Plot, Save, and Spec columns are saved for importing or exporting. However, if you have specified values in other columns as well, all other columns, such as Notation, are also saved in the CSV file.
- Outputs that were earlier exported from the ADE Explorer environment in a `.csv` file can be imported in ADE Explorer.


© 2023 Cadence Design Systems, Inc.

# Specifying which Outputs to Save

Saving all the node voltages and terminal currents for a large design produces an enormous data set. The analog circuit design environment lets you save a selected set of voltages and currents from the schematic.

Once you select a set of output nodes and terminals, you can save their names to a file using the *Save State* command. You do not need to explicitly save nets and terminals that are used in expressions. All nets and nodes that are used in expressions are automatically set for saving and are also added to the Outputs Setup tab.

After you select the outputs you want to save, the next step is to start the simulation.

This topic covers the following sub-topics:

- Saving Selected Voltages or Currents
- Saving All Voltages or Currents

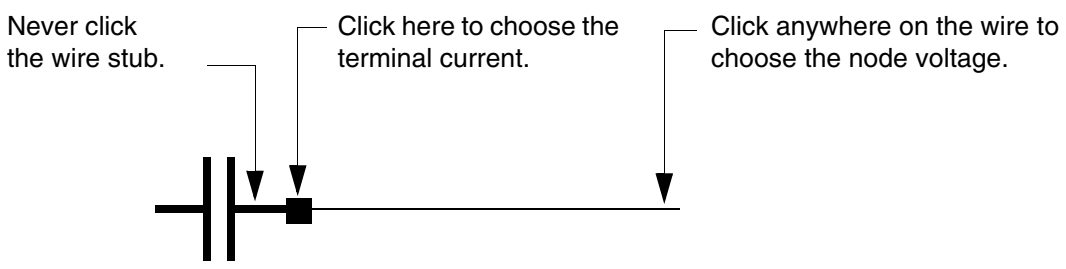
## Saving Selected Voltages or Currents

To save the simulation data for particular nodes and terminals,

1. Choose *Outputs – To Be Saved – Select On Design*.
2. In the Schematic window, choose one or more nodes or terminals.

   The system circles pins when you choose a current and highlights wires when you choose a net.

   - Click an instance to choose all instance terminals.
   - Click the square pin symbols to choose currents.
   - Click wires to choose voltages.
   - Click and drag to choose voltages by area.





© 2023 Cadence Design Systems, Inc.

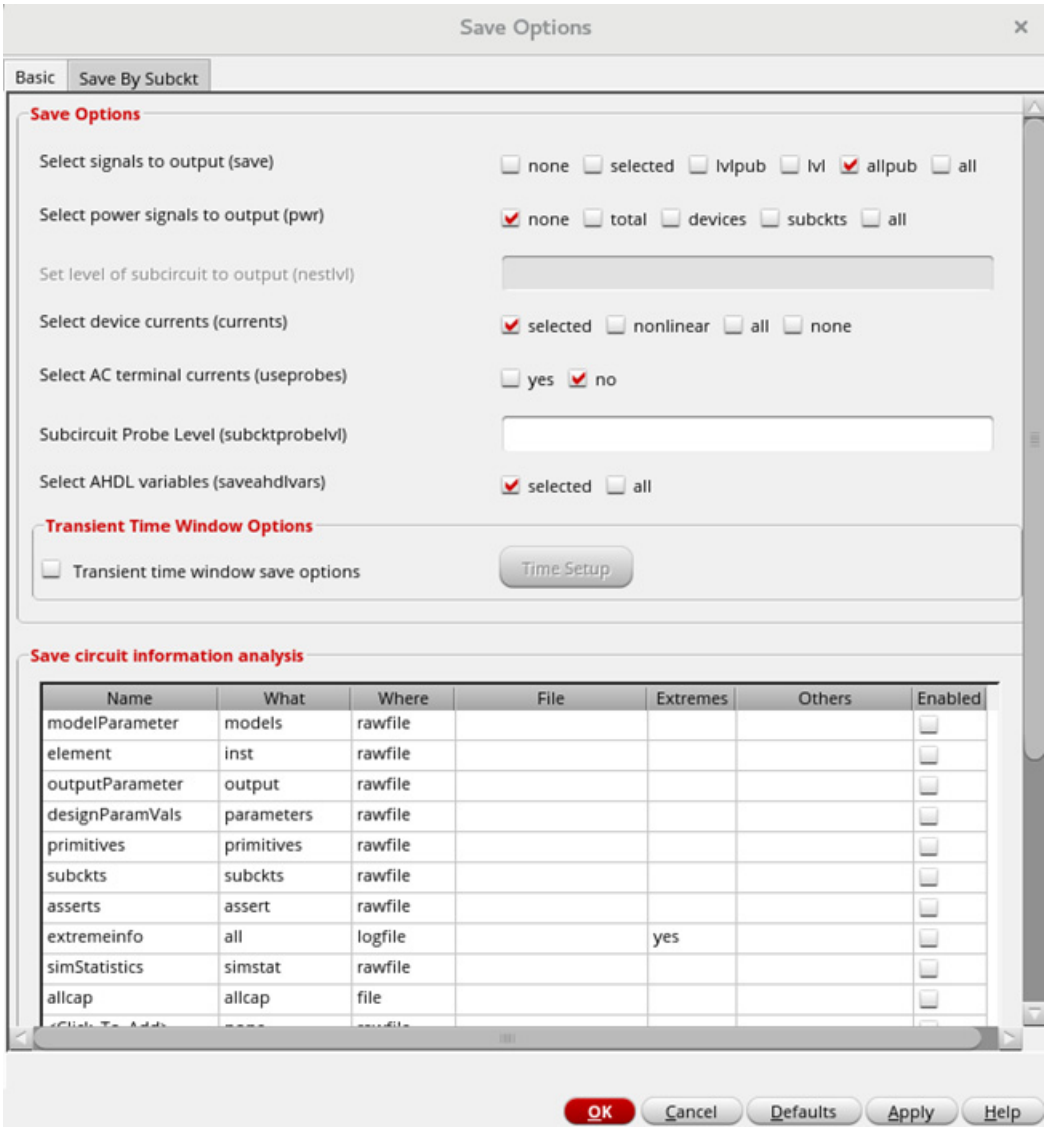
Press the `Esc` key when you finish.

## Saving All Voltages or Currents

To save all of the node voltages and terminal currents,

1. Choose *Outputs – Save All*.

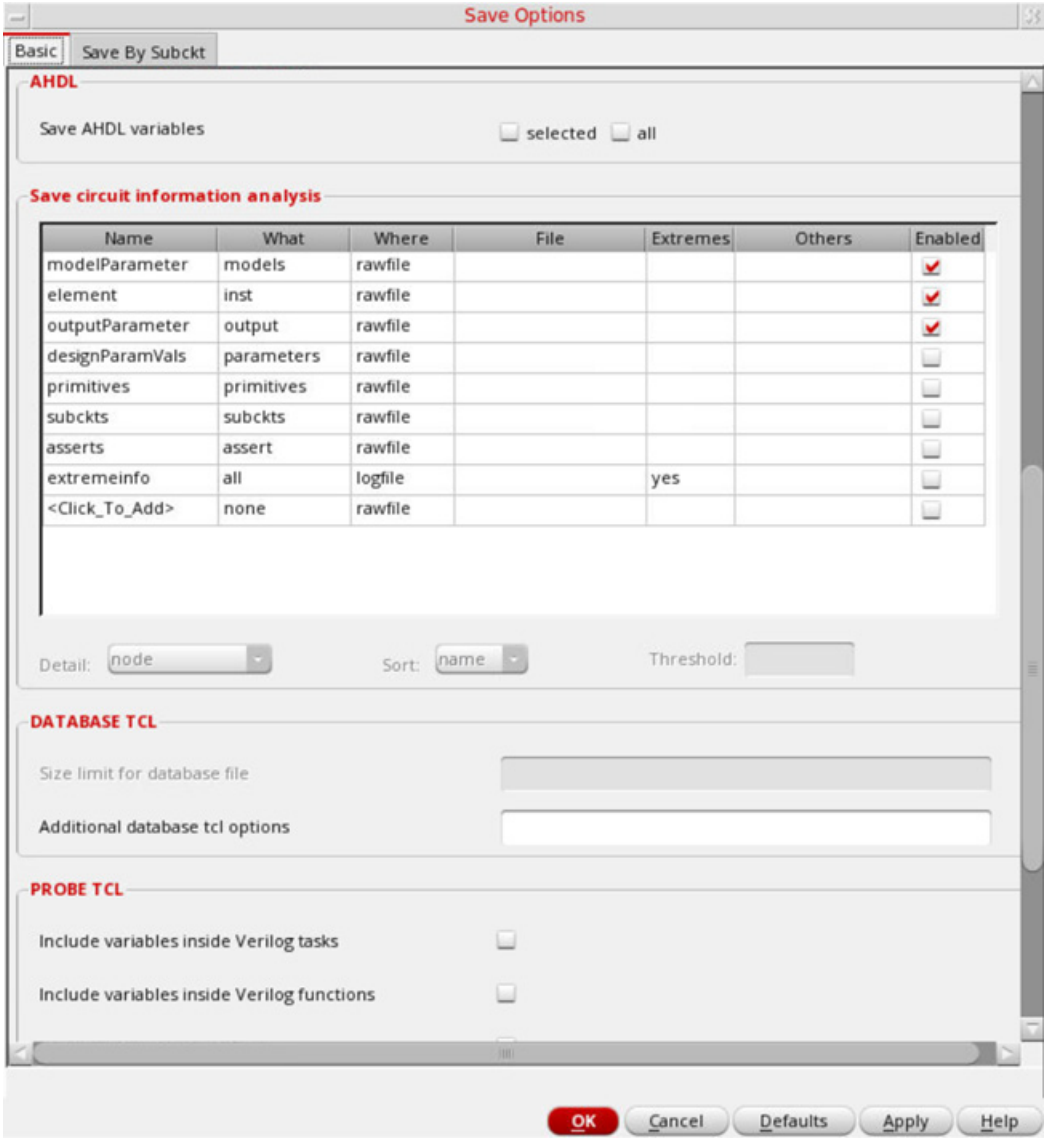
   A form appears that varies according to the simulator you use. For example, if you use the Spectre simulator, the Save Options form appears with the following format.





© 2023 Cadence Design Systems, Inc.

When you use the AMS simulator, the following Save Options form appears.




With AMS, the *Save All* command is not on by default. Save (or probe) statements are placed in the `probe.tcl` file and not in the `amsControl.scs` file. For information about this, see the “Tcl-Based Debugging” appendix chapter of *Spectre AMS Designer and Xcelium Simulator Mixed-Signal User Guide.*

1. Select the values you want to save and click *OK*.

**Note:** For AMSUltra, the options *Save model parameters info*, *Save elements info*, and *Save output parameters info* appear deselected by default. If you select them, the speed with which AMSUltra typically works may be compromised especially if you have a big design.


© 2023 Cadence Design Systems, Inc.

All of the cross-references are to sections in the *Specifying Output Options* chapter of the *Spectre Classic Simulator, Spectre APS, Spectre X, and Spectre XPS User Guide*.

The table below describes the fields of the *Basic* tab in *Save Options* form:

| Field | For more information, see |
|---|---|
| Select signals to output (save) | *The save Parameter Options* in the *Spectre Classic Simulator, Spectre APS, Spectre X, and Spectre XPS User Guide.* |
| Select power signals to output (pwr) | *Saving Power* in the *Spectre Classic Simulator, Spectre APS, Spectre X, and Spectre XPS User Guide.* |
| Set level of subcircuit to output (nestlvl) | *Saving Groups of Signals* in the *Spectre Classic Simulator, Spectre APS, Spectre X, and Spectre XPS User Guide*. |
| Select device currents (currents) | *Saving Groups of Currents* in the *Spectre Classic Simulator, Spectre APS, Spectre X, and Spectre XPS User Guide.* |
| Select AC terminal currents (useprobes) | *Setting Multiple Current Probes* in the *Spectre Classic Simulator, Spectre APS, Spectre X, and Spectre XPS User Guide.* |
| Set Subcircuit probe level (subcktprobelvl) | *Saving Subcircuit Terminal Currents* in the *Spectre Classic Simulator, Spectre APS, Spectre X, and Spectre XPS User Guide* |
| Select AHDL variables (saveahdlvars) | *Saving All AHDL Variables* in the *Spectre Classic Simulator, Spectre APS, Spectre X, and Spectre XPS User Guide.* |

***The** all **buttons (**Select signals to output (save)* ***) are global buttons but can be locally overridden while specifying options for a particular analysis.***

Spectre never saves the current data for global pins even if you select Save all on the *Outputs* tab.


© 2023 Cadence Design Systems, Inc.

### Specifying Hierarchy Levels to Save Outputs

When the circuit has multiple levels of hierarchy, saving the voltages and currents for a subcircuit can produce an enormous data set, resulting in degraded simulator performance or out-of-memory problems. To address this issue, you can set the level of hierarchy to which you want to save voltages, currents and power signals from the schematic.

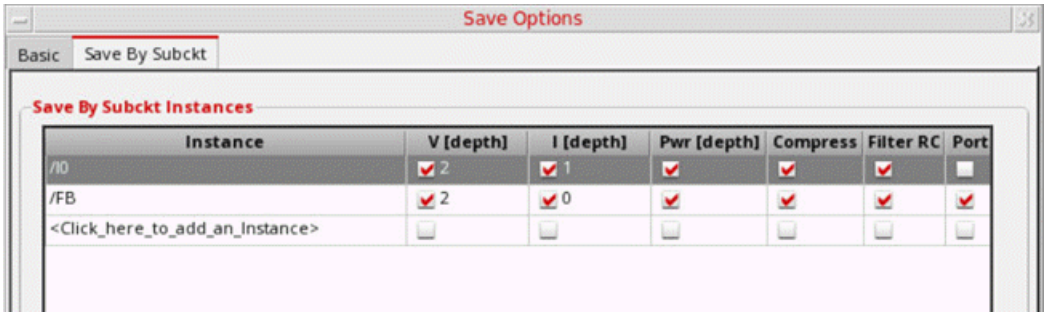
*Caution*

***Any changes made in the** Save By Subckt Instances **table on the Save Options form are saved and reflected in the netlist even if you click** Cancel **on the form. If you do not want to save the changes in this table, delete the rows or edited entries.***

To do this, you can use the *Save By Subckt* tab on the *Save Options* form.

- Select the check box for voltage `V(depth)`, current `I(depth)`, or power `pwr (depth)` that you want to save.
- Specify the hierarchy level to which you want to save outputs next to the check box.

  In the following figure, for the instance `/I0`, the voltage for two levels and the current for one level of hierarchy is saved.




  **Note:** If you do not specify the hierarchy level to save, the voltage, current, and power for all the levels in the subcircuit are saved.

- Select *Compress* if you want to reduce the size of the output file. The simulator then saves the data for a signal only when the value of that signal changes.
- Select *Filter RC* to filter out the nodes that are connected only to parasitic elements from the output signal list.
- Select *Ports* if you want to save the output port information for the specified subcircuits.
- Select *Exclude* to specify the list of instances to be excluded during simulation.


© 2023 Cadence Design Systems, Inc.

- From the *sigtype* drop-down list, select the signal type.

  The available options are

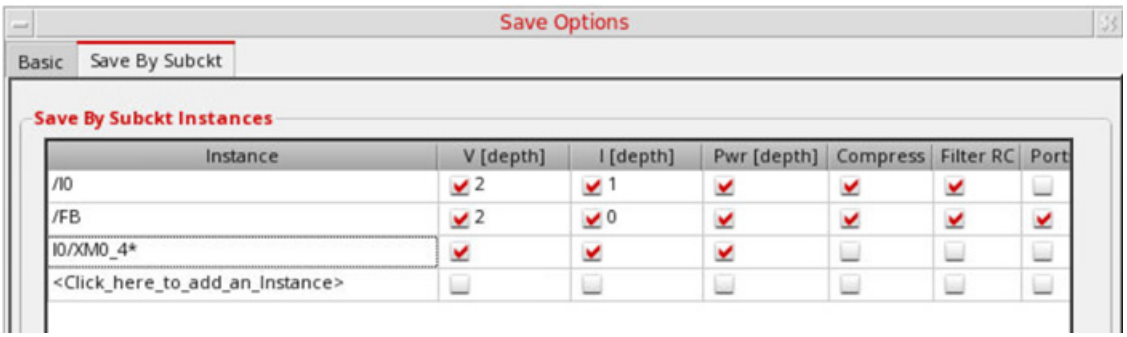
  - *node*: saves node voltages for all instances of the specified subcircuit.
  - *dev*: saves information for all devices contained in the instances of the subcircuit.
  - *subckt*: saves all the terminal currents of the specified subcircuit.
  - *all*: saves all node voltages, subcircuit terminal currents and all available device information including terminal currents and operating point parameters. This is the default selection.

- Specify the additional save options in the *User Options* field. For example, `spectre -h save`
- In the User Probe Options field, specify the additional save options for AMS. These options are then saved to the `probe.tcl` file

  **Note:** This option is available only for AMS simulator.

*Tip*

The *Instance* column in the *Save By Subckt Instances* table supports wildcards, which makes it easier to save the correct signals down the hierarchy. For example, to save only the voltage, current and power signals under `I0/XM0_4` add the syntax `I0/XM0_4*.`



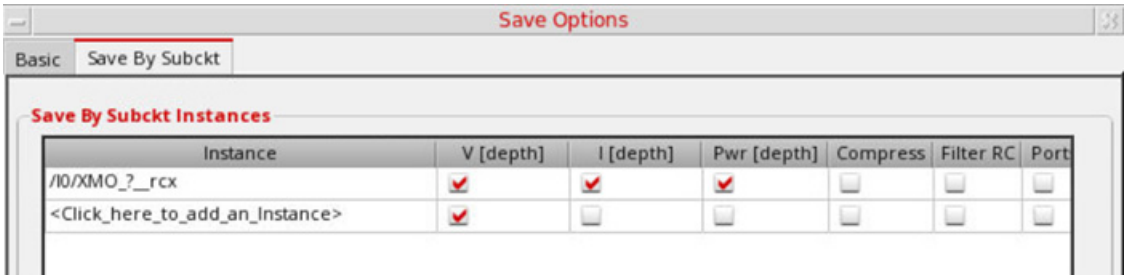

The resultant netlist contains the following save statement:

```
save I0.XM0_4*.*:currents sigtype=all
save I0.XM0_4*.*
save I0.XM0_4*.*:pwr sigtype=all
```

The Results browser also contains signals in `IO`, starting with the string `XM0_4`.


© 2023 Cadence Design Systems, Inc.

Another example, to match the instances between `XM0_1__rcx` and `XM0_9__rcx` only, use the wildcard `?`. It is used to match only one character.




### Saving Pcell Operating Point Parameters

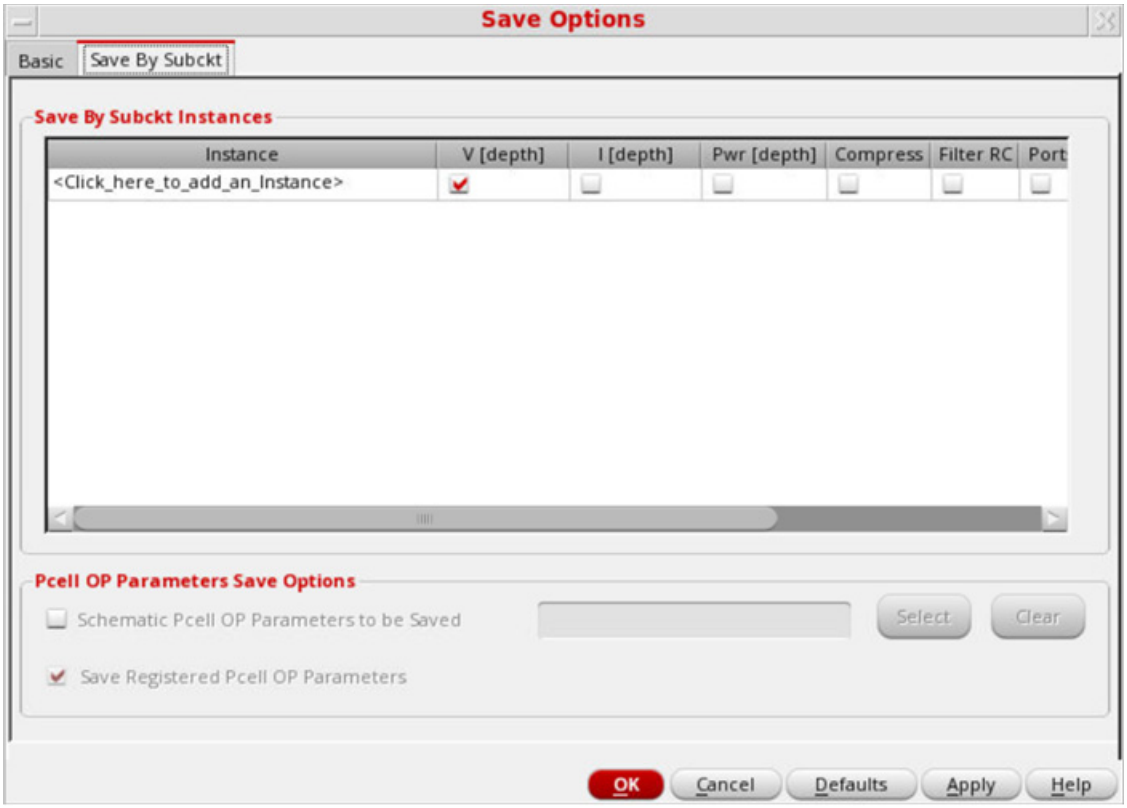
The *Save By Subckt* tab on the *Save Options* form is used to manage Subckt instances and Pcell.

To save the operating point parameters of a selected Pcell:

1. Select the *Schematic Pcell OP Parameters to be Saved* check box. You can either type the names of the OP parameters in the text box, or click *Select* to choose the operating point parameters of the specified Pcell from the schematic. When you choose from schematic, the text box displays the selected operating points as `"<name-ofoppoint>:oppoint"` and the netlist is generated in the format:


© 2023 Cadence Design Systems, Inc.

```
save NM0:oppoint M0:oppoint devtype=subckt
```




2. Select the *Save Registered Pcell OP Parameters* check box to save the operating points of Pcells registered using the function sevRegisterPcellsForAnnotation. This check box is enabled only if *Schematic Pcell OP Parameters to be Saved* is selected.

   For example, if you specify the Pcell in the format `"<lib-name>/<cell-name>"`, the netlist is generated as:

   ```
   save *:oppoint devtype=nmos1v_s_pcell_3
   ```

   If you specify the Pcell in the format `"<lib-name>/<cell-name> <commaseparated-list-of-parameters>"`, the netlist is generated as:

   ```
   save *:vth *:id devtype=nmos2v_s_pcell_1

   save *:vth *:id devtype=nmos2v_s_pcell_2

   save *:vth *:id devtype=nmos2v_s_pcell_0
   ```

**Note:** To set the check box *Schematic Pcell OP Parameters to be Saved* as the default selection, set the environment variables `subcktoppoint` and `savePcell` in your `.cdsenv` file.

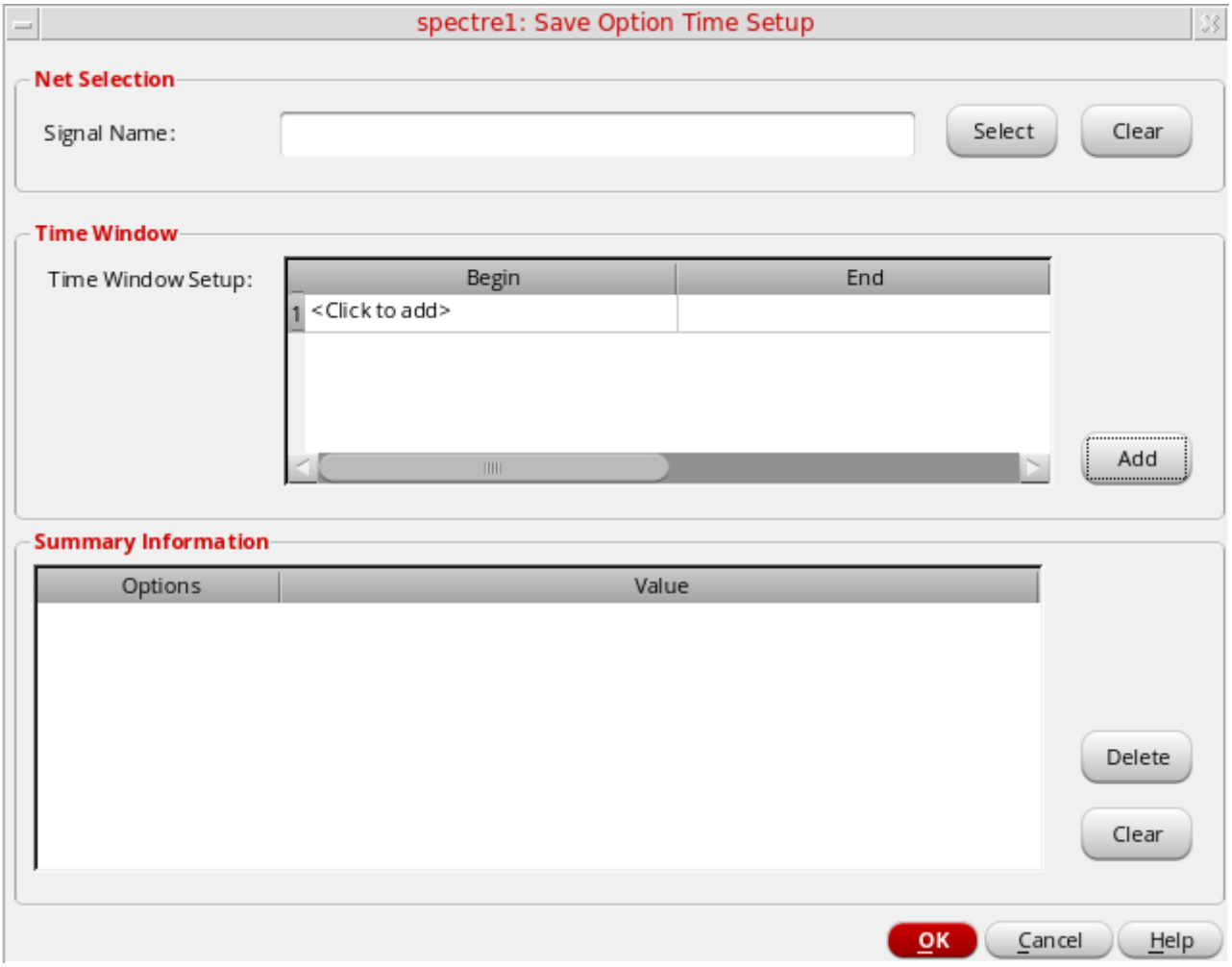

© 2023 Cadence Design Systems, Inc.

### Saving Transient Time Window Setup

To save the time interval or time window for which you need to save outputs for signals during transient analysis:

1. In the *Transient Time Windows Options* section, select the *Transient time window save options* check box in the *Basic* tab of *Save Options* form. By default, this check box is disabled.
2. Click the *Time Setup* button.

   The Save Option Time setup form is displayed. By default, the output data for signals for the complete simulation time is saved. This form helps you specify the time windows for specific signals.




This form contains the following sections:

- *Net Selection*


© 2023 Cadence Design Systems, Inc.

- *Signal Name* — The nodes or signals. Click *Select* to open the schematic view in a new window, where you can select the nodes or signals. Click *Clear* to clear *Signal Name*.

**Note:** You can specify a list of either signals or nodes, or a combination of both when selecting the Signal name from the schematic, to save the output data for the same time window for different signals or nodes

- *Time Window*
  - *Time Window Setup* — Sets up one or more time windows for which you need to save the output data for the node or signal specified in *Signal Name*. Click *Add* to add these time windows to the *Summary Information* table. For each time window you can specify the *Begin* and *End* nodes in the respective columns.
- *Summary Information* — Used to display the summary of the signals and their corresponding time windows. Click *Delete* to remove the time window details for the


© 2023 Cadence Design Systems, Inc.

specified node or signal from the netlist. Click *Clear* to clear the Summary Information table.

The netlist file saves the Summary Information in the following format:

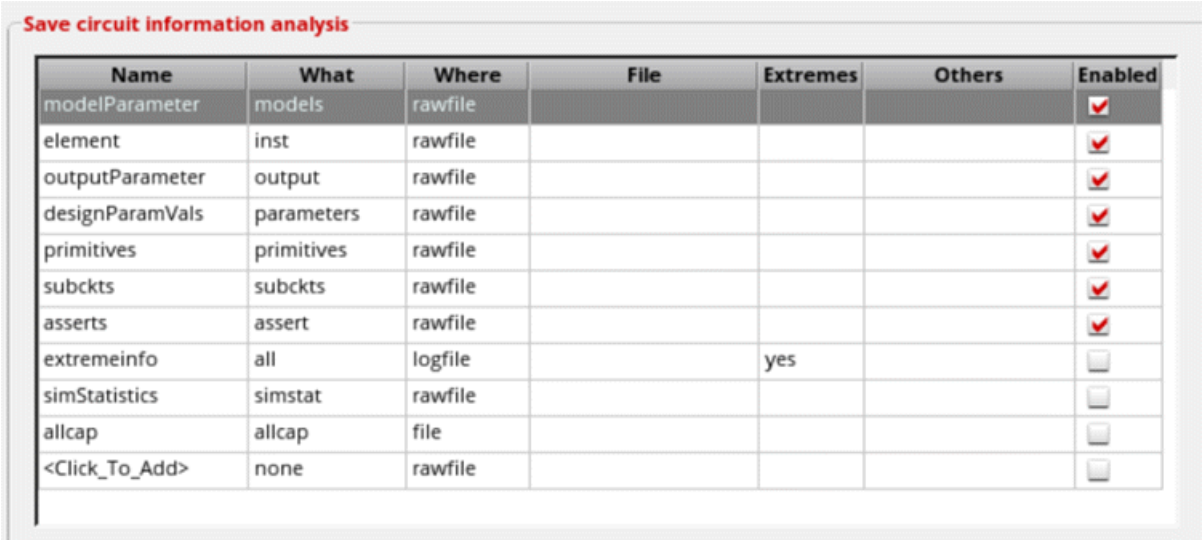
```
save N1 time_window=[1 2]
save IO.* time_window=[1n 2n]
save IO.* time_window=[1n 2n 3n 4n]
```

**Note:** Nodes or signals that start with the same name can be referred to by adding a wildcard. Here, save `IO.* time_window=[1n 2n]` implies that the output data is to be saved for the time window `[1n 2n]` for all the signal names starting with "`IO`".


© 2023 Cadence Design Systems, Inc.

### Saving Circuit Information Analysis

You can use the *Save circuit information analysis* table in the *Basic* tab of *Save Options* form to save the information about the circuit and its components. The table data can be saved in a state file and loaded into another session. To save the table data in a state file, select the *Outputs* check box in the S*aving State* form.




| Name | What | Where | File | Extremes | Others | Enabled |
|---|---|---|---|---|---|---|
| modelParameter | models | rawfile | | | | ☑ |
| element | inst | rawfile | | | | ☑ |
| outputParameter | output | rawfile | | | | ☑ |
| designParamVals | parameters | rawfile | | | | ☑ |
| primitives | primitives | rawfile | | | | ☑ |
| subckts | subckts | rawfile | | | | ☑ |
| asserts | assert | rawfile | | | | ☑ |
| extremeinfo | all | logfile | | yes | | ☐ |
| simStatistics | simstat | rawfile | | | | ☐ |
| allcap | allcap | file | | | | ☐ |
| <Click_To_Add> | none | rawfile | | | | ☐ |

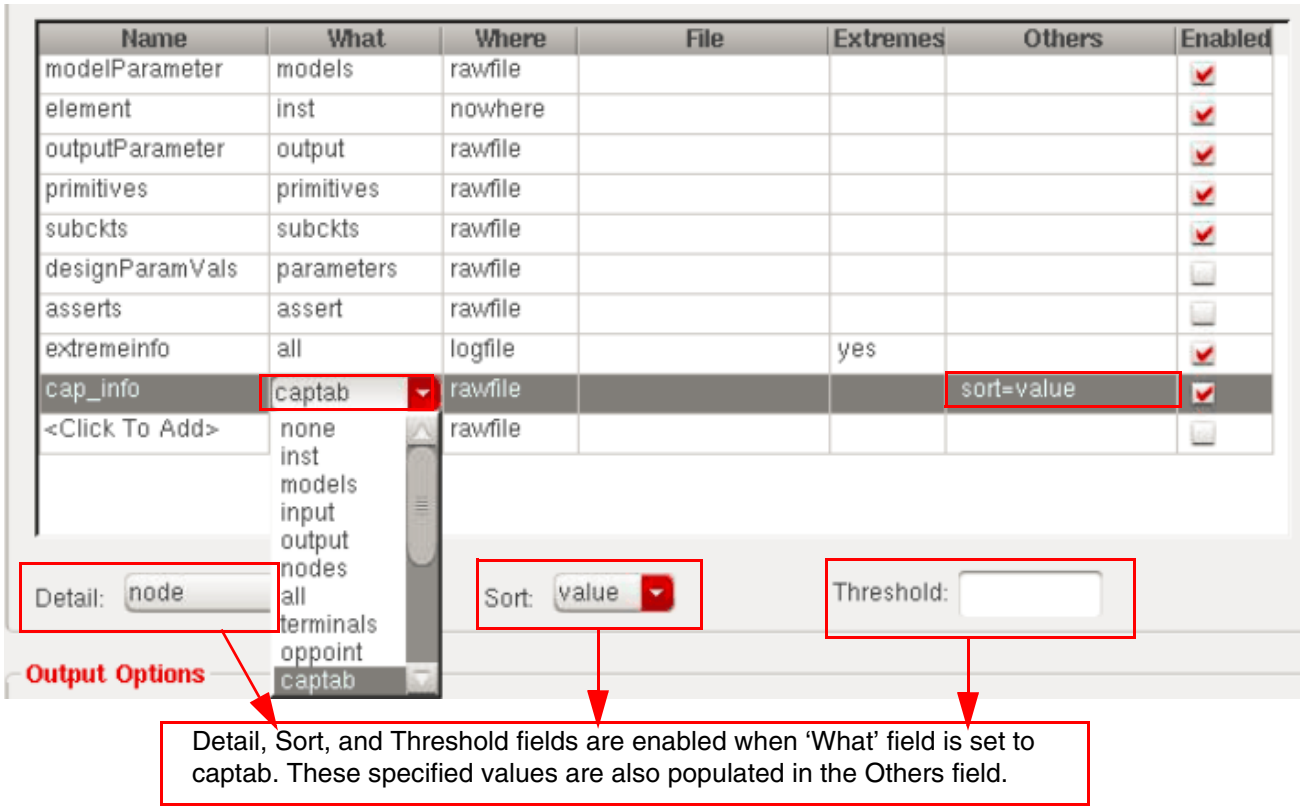
This table includes the following fields:

- *Name*—Name of the information analysis.
- *What*—The parameters that are to be saved.

  Possible values are `none`, `inst`, `models`, `input`, `output`, `nodes`, `all`, `terminals`, `oppoint`, `captab`, `parameters`, `primitives`, `subckts`, `assert`, `allparameters`, `netlist`, `options`, and `dumpall`.
- *Where*—File in which the parameters are to be saved. Asserts can be saved only in a rawfile.
  Possible values are `nowhere`, `file`, `logfile`, and `rawfile`.
- *File*—When *Where* is set to `file`, this field is used to specify the name of the file in which parameters are to be saved.
- *Extremes*—Specifies whether the minimum or maximum values are to be saved.
  Possible values are `no`, `yes`, and `only`.
  When this field is set to `no`, the original information, such as `opinfo`, is saved.
  When set to `yes`, the original information and minimum or maximum values are saved.
  When set to `only`, the minimum or maximum values are saved.


© 2023 Cadence Design Systems, Inc.

- *Others*—When the *What* field is set to `captab`, the options located at the bottom of the table become active and the values specified in the *Sort* and *Threshold* fields are populated in the *Others* field (see the figure below).
  - *Detail*—Specifies the level of detail for the capacitance table. Possible values are `node`, `nodetoground`, and `nodetonode`.
  - *Sort*—Sorts the capacitance table. Possible values are `name` and `value`.
  - *Threshold*—Specifies the threshold value to be used for saving capacitance. A capacitance value smaller than the threshold value is ignored.




- *Enabled*—Select this check box to save the corresponding parameter.

To add a new parameter to the table, double-click `<Click_To_Add>` and specify the required values in various fields.

To delete a parameter, right-click anywhere in the row that contains the parameter and choose *Delete*.

**Note:** You can only delete the parameters that you added to the table manually.


© 2023 Cadence Design Systems, Inc.

For more information, see Control Statements in the *Spectre Classic Simulator, Spectre APS, Spectre X, and Spectre XPS User Guide.*

### Setting Up Output Options

The table below describes the *Output Options* field of the Save Options form. These options specify the format in which the results data must be saved.

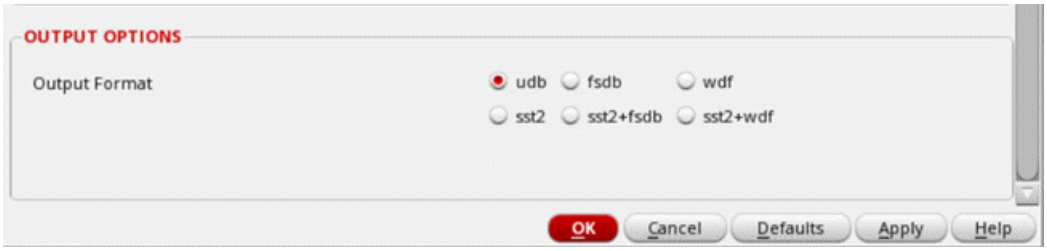
On selecting Spectre as the simulator, the following output formats are displayed.


© 2023 Cadence Design Systems, Inc.

| Field | Description |
|---|---|
| Output Format | Specifies the format in which the results data must be saved. The results data can be saved in the following formats:<br>■ `sst2` – Signal Scan 2 (SST2) format. This format is supported for transient analyses only. Non-transient data cannot be generated in the SST2 format and is generated in parameter storage format (PSF) format.<br>■ `psf` – Cadence parameter storage format (PSF)<br>■ `psf with floats` – Cadence lowered precision parameter storage format.<br>**Note:** The `psf with floats` format stores data using floats versus doubles. This results in a database that is about 50% of the size of normal PSF output, thereby providing capacity and performance benefits. For some simulators, using this format is not recommended as the use of floats reduces the number of significant digits and can limit the accuracy of measurements. Specifically, SpectreRF users are recommended not to use this format as the noise floor numbers can be misleading.<br>■ `psfxl` – Cadence parameter storage XL format (PSF XL). PSF XL provides higher performance for large circuit designs.<br>**Note:** When `psfxl` is set as the output format for transient analysis, the behavior of the waveform writing module is controlled by a set of environment variables. For more details, refer to Environment Variables for PSFXL Output Format.<br>*Caution*<br>***The PSFXL format is not compatible with versions of Virtuoso Visualization and Analysis prior to IC6.1.4 unless the CDS_PSFXL_COMPAT environment variable is defined by the user. For information on this variable, refer to*** Table 4-2 ***on page 490.***<br>■ `fsdb` – Fast Signal Database format (FSDB). |


© 2023 Cadence Design Systems, Inc.

On selecting AMS as the simulator, the following output formats are displayed.




| Field (Output Format) | Description |
|---|---|
| `udb` | Unified Database (UDB) format. On using this format, the analog signals are saved in `psfxl` format while the digital signals are saved in `sst2` format. When the results with `udb` format are viewed using Virtuoso Visualization XL or SimVision, both analog and digital signals are displayed in one dataset. |
| `fsdb` | Fast Signal Database (FSDB) format.<br>**Note:** This format is supported only for transient analyses. |
| `wdf` | Waveform Data Format (WDF).<br>**Note:** This format is supported only for transient analyses. |
| `sst2` | Signal Scan 2 (SST2) format. This format is supported only for transient analyses. Non-transient data cannot be generated in the SST2 format and is generated in Parameter Storage Format (PSF). |
| `sst2+fsdb` | Signal Scan 2 (SST2) format and Fast Signal Database (FSDB) format.<br>On selecting this output format, the analog and digital signals are saved as follows.<br>■ Analog signals are saved in `sst2` and `fsdb` format.<br>■ Digital signals are saved in `sst2` and `vcd` format. |


© 2023 Cadence Design Systems, Inc.

| Field (Output Format) | Description |
|---|---|
| `sst2+wdf` | Signal Scan 2 (SST2) format and Waveform Data Format (WDF).<br>On selecting this output format, the analog and digital signals are saved as follows.<br>■ Analog signals are saved in `sst2` and `wdf` format.<br>■ Digital signals are saved in `wdf` format and `vcd` format. |


© 2023 Cadence Design Systems, Inc.

# Expressions to be Measured Across Dimensions

Simulation results data is spread across multiple dimensions—sweeps or corners. Therefore, at times, it is required to measure outputs across these dimensions. You can do this by specifying the dimension across which you want to evaluate a result.

ADE Explorer provides the following ways to evaluate expressions across different dimensions:

Measurements across dimensions are supported only in the following run modes:

- Single Run, Sweeps, and Corners
- Monte Carlo

***Related Topics***

- Measuring Outputs Across Corners
- Measuring Outputs Across Sweeps or Across All
- Creating Expressions to be Measured Across Any Dimension


© 2023 Cadence Design Systems, Inc.

# Environment Variables for PSFXL Output Format

When `psfxl` is set as the output format for transient analysis, the behavior of the waveform writer module is controlled by a set of environment variables defined in your shell session. These variables are listed in Table 4-2 on page 490. You can override the default values of these variables by setting them in your .cshrc or in your shell session.

*Important*

The default values of the variables given in Table 4-2 on page 490 have been set to provide optimal performance for most end-user designs. Users should not change the default values unless they experience specific performance degradation issues.

**Table 4-2 Environment Variables for PSFXL Output Format**

| Variables | Description |
|---|---|
| CDS_PSFXL_FLUSH_INTERVAL | Specifies the number of CPU seconds elapsed between data flushes to disk. This value is increased by 2% after every flush upto the value of variable CDS_PSFXL_MAX_FLUSH_INTERVAL.<br><br>Default value is `300` seconds.<br><br>Larger flush intervals require more simulation data to be kept in memory, but results in more efficient reading of data by the waveform and post-processing tools. Smaller intervals reduce the memory usage and runtime overhead of the simulator. |
| CDS_PSFXL_INIT_FLUSH_INTERVAL | Specifies the rate (in terms of number of CPU seconds elapsed) at which data is initially flushed to the disk. This value is progressively increased by 50% after each flush until it approaches the value of variable CDS_PSFXL_FLUSH_INTERVAL. After this point, the flush rate is determined by the behavior of the variable CDS_PSFXL_FLUSH_INTERVAL.<br><br>Default value is `60` seconds. |


© 2023 Cadence Design Systems, Inc.

| Variables | Description |
|---|---|
| CDS_PSFXL_MAX_FLUSH_INTERVAL | Specifies the maximum number of CPU seconds elapsed between data flushes. This sets an upper limit on how large the progressively increasing flush period defined by CDS_PSFXL_FLUSH_INTERVAL can grow.<br><br>Default value is `1200` seconds.<br><br>Larger flush intervals require more simulation data to be kept in memory, but results in more efficient reading of data by the waveform and post-processing tools. Smaller intervals reduce the memory usage and runtime overhead of the simulator. |
| CDS_PSFXL_MAX_BUFSIZE | Specifies how much in-process memory to be allocated for unwritten data values.<br><br>Default value is `256` MB for 32-bit platform and `384` MB for 64-bit platform.<br><br>You need to specify the value of this variable in Bytes. For example, if you need to specify the `CDS_PSFXL_MAX_BUFSIZE` to `1024` MB, you need to set `CDS_PSFXL_MAX_BUFSIZE` as follows:<br><br>`setenv CDS_PSFXL_MAX_BUFSIZE 1073741824`<br><br>Larger buffer size increase the performance of the viewer or reader module. Smaller buffers reduce the memory usage and runtime overhead of the simulator. |
| CDS_PSFXL_COMPAT | Instructs the simulator to write both PSF and PSFXL data simultaneously, thereby preserving backward compatibility for users of Virtuoso Visualization and Analysis prior to IC6.1.4.<br><br>Default value `0` disables this compatibility mode. Setting the value to `1` preserves backward compatibility, but requires twice as much disk space to be consumed, and leads to slightly higher runtime overhead on the simulator. |


© 2023 Cadence Design Systems, Inc.

| Variables | Description |
|---|---|
| CDS_PSFXL_FLOATY | Sets the default precision of waveform Y-values. This has no effect on the X- or time sweep values that are always saved as double precision. By setting this variable, you can reduce the size of waveform database by approximately 50% for spectre and aps simulators and approximately 25% for ultrasim. This variable has negligible impact on simulations and waveform viewing.<br><br>Default value `0` sets the default precision of waveform Y-values to double precision float. Set this variable to `1` to set the default precision to single precision float.<br><br>*Important*<br><br>The `CDS_PSFXL_FLOATY` environment variable will be removed in a future release. It is recommended that you use the Spectre command-line option `uwiprecision` in its place. |


© 2023 Cadence Design Systems, Inc.

| Variables | Description |
|---|---|
| CDS_PSFXL_SERVER | Enables or disables reading of live simulation data while the simulation is running. Live simulation data includes both of the following:<br>■ the simulation data that has been flushed to the disk<br>■ data that is still in the simulation process memory and not yet flushed to the disk<br>By enabling the reading of live simulation data, you can improve the performance of waveform viewing because it is not required to read the entire waveform data every time the data is flushed by the simulator. This has negligible impact on simulation unless excess live simulation data reads are requested by you. Therefore, to maximize the performance of the simulator, it is recommended to minimize the amount of live data requests for plotting while the simulation is in progress.<br>Default value `0` disables reading of live simulation data. Set this variable to a non-zero integer value to enable reading. |

## PSFXL Environment Variables for Site Administrators

In addition to the environment variables given in Table 4-2 on page 490, there are two more variables that control how the live simulation data in PSFXL format is read. These variables define the port range for server connections and are intended for site administrators who want to enforce network service restrictions across their IT domain. The site administrators can set these variables in their shell session. Any value set for these variables in users local home is ignored.

**Table 4-3 Environment Variables for Site Administrators**

| | |
|---|---|
| CDS_PSFXL_PORT_LOW | Lowest TCP/IP port number allowed for server connections. The server will not search for any open ports below this value.<br>Default value: `32768` |


© 2023 Cadence Design Systems, Inc.

**Table 4-3 Environment Variables for Site Administrators**

| | |
|---|---|
| CDS_PSFXL_PORT_HIGH | Highest TCP/IP port number allowed for server connections. The server will not search for any open ports above this value.<br><br>Default value: `65536` |

Setting both the variables given in the previous table to `0` disables communications with the simulator for reading live PSFXL data. It is recommended that site administrators use the default values set for these variables. Making the port range too restrictive might limit the number of live simulations that can be made available to the end user. For example, if the range is set to 38000 to 38009, then only ten simulation processes can be used simultaneously by the users.


© 2023 Cadence Design Systems, Inc.

# 5

# Working with Device Instance Parameters

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

This chapter covers the following topics:

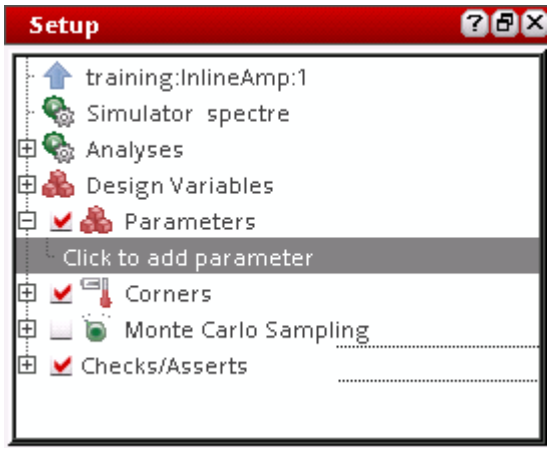
- Creating Device Instance Parameters on page 496
- Changing the Value of Device Instance Parameters on page 504
- Specifying Values of Device Instance Parameters on page 506
- Creating Matched Device Parameters on page 506
- Creating Ratio-Matched Device Parameters on page 508
- Deleting a Device Parameter on page 510
- Enabling and Disabling Parameters on page 512
- Adding Notes to Parameters on page 512
- Viewing Parameters by Property and Object Names on page 514
- Filtering Device Instance Parameters on page 515
- Creating Custom Device Filters on page 517
- Changing the Order of Device Parameters on page 518
- Disabling Callbacks on Swept Device Parameters on page 519
- Adding or Modifying Specifications for Parameters and Design Variables on page 520
- Creating Parameter Ranges on page 531
- Working with Parametric Sets on page 535


© 2023 Cadence Design Systems, Inc.

# Creating Device Instance Parameters

Perform the following steps to create instance parameters for the design associated with your test:

1. In the *Parameters* section on the *Setup* assistant, click *Click to add parameter*.

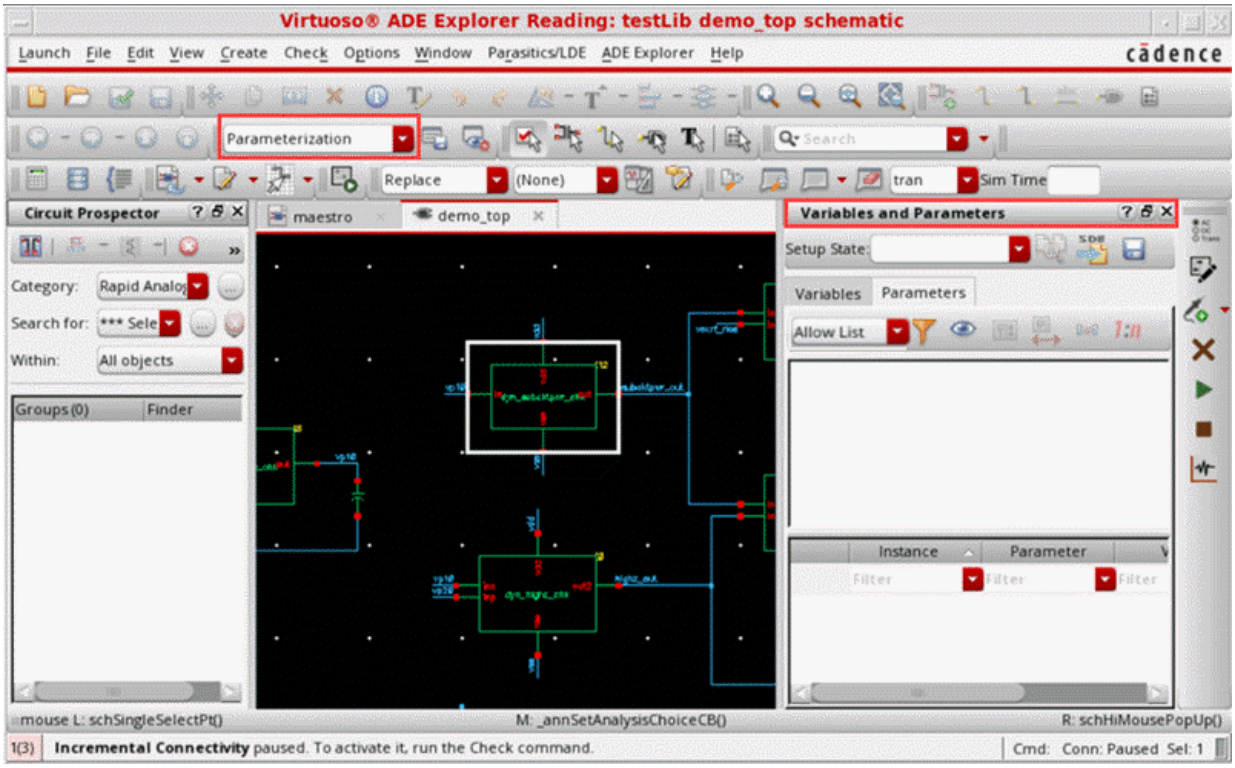



   Alternatively, right-click *Parameters* and choose *Add Parameter*.

   The schematic associated with the test is displayed in a new tab in the ADE Explorer window. The menu changes to that of *Virtuoso Schematic Editor* and the `Parameterization` workspace is set.


© 2023 Cadence Design Systems, Inc.

In this workspace, the *Variables and Parameters* assistant is displayed. The *Parameters* tab on the *Variables and Parameters* assistant is also opened.

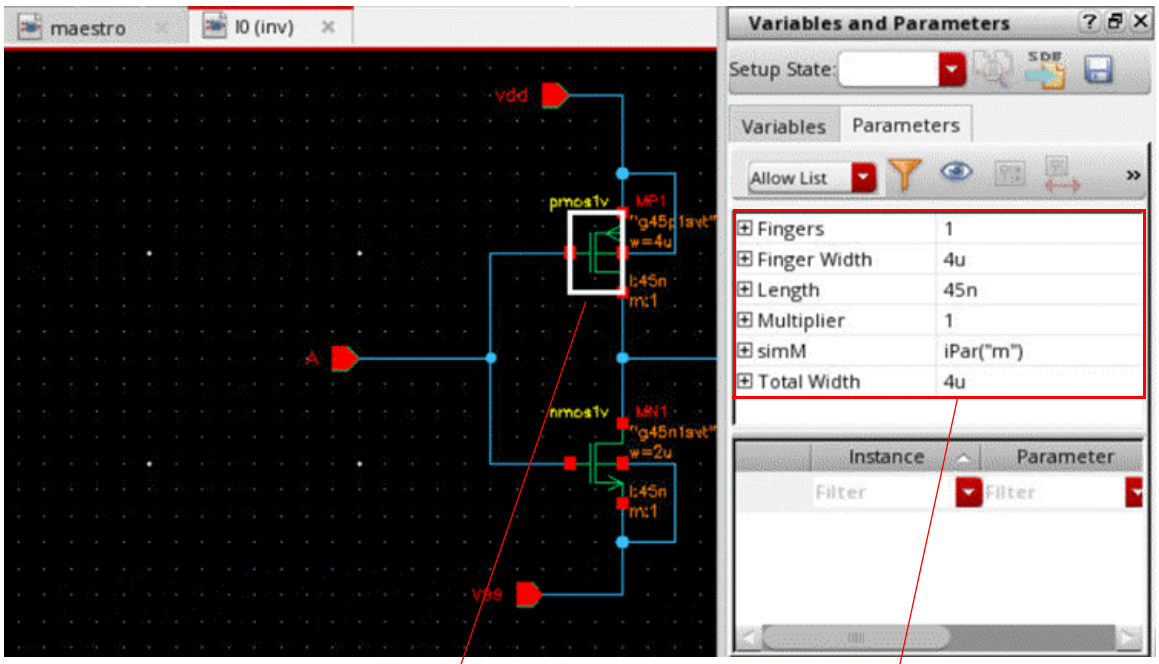



2. On the schematic, select one or more device instances.

   To select more than one instance, hold down the *Shift* key and click each instance one by one or by dragging a selection box around a set of instances.


© 2023 Cadence Design Systems, Inc.

The parameter properties of the selected devices are listed in the upper half of the *Parameters* tab of the *Variables and Parameters* assistant.




Selected device instance

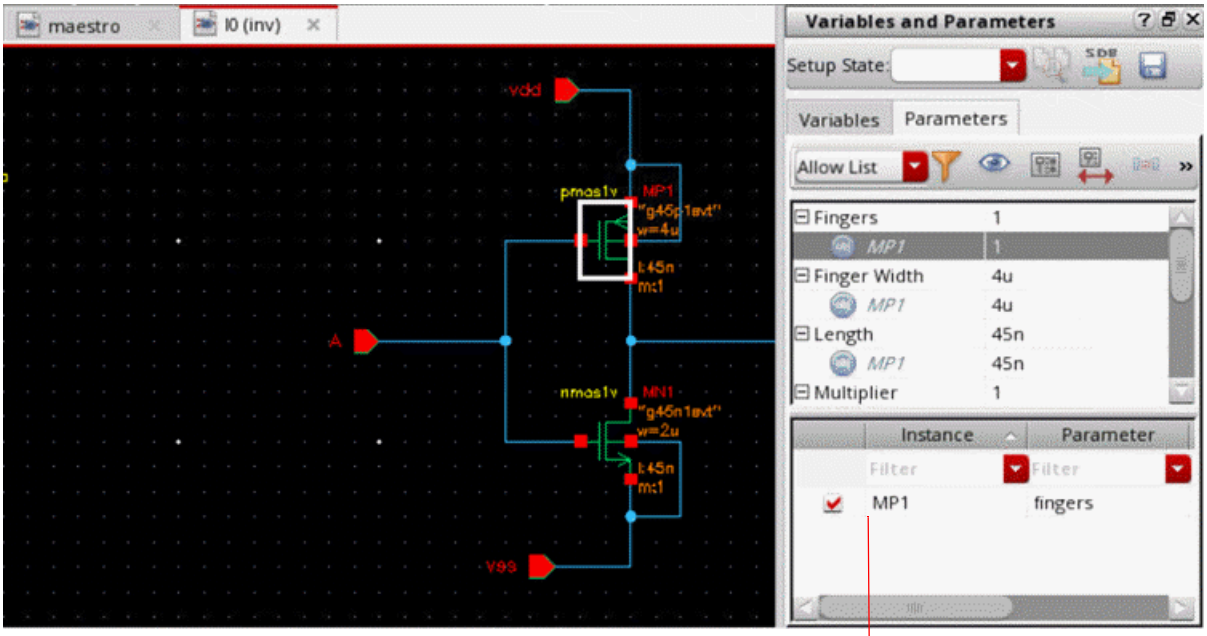
Parameter properties of the selected instance.

3. Expand a property section to view the device names.
4. Right-click an instance name and choose *Create Parameter*.

   Alternatively, click *Create Parameter* on the toolbar.

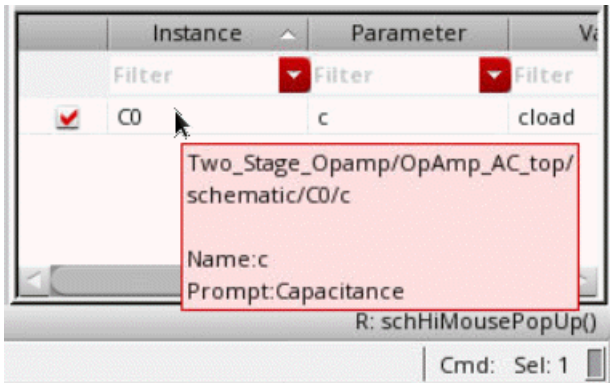

© 2023 Cadence Design Systems, Inc.

A parameter is created for the selected property of the selected device instance. The details of the added parameter appear in the lower half of the *Variables and Parameters* assistant.




The newly created parameter is added here.

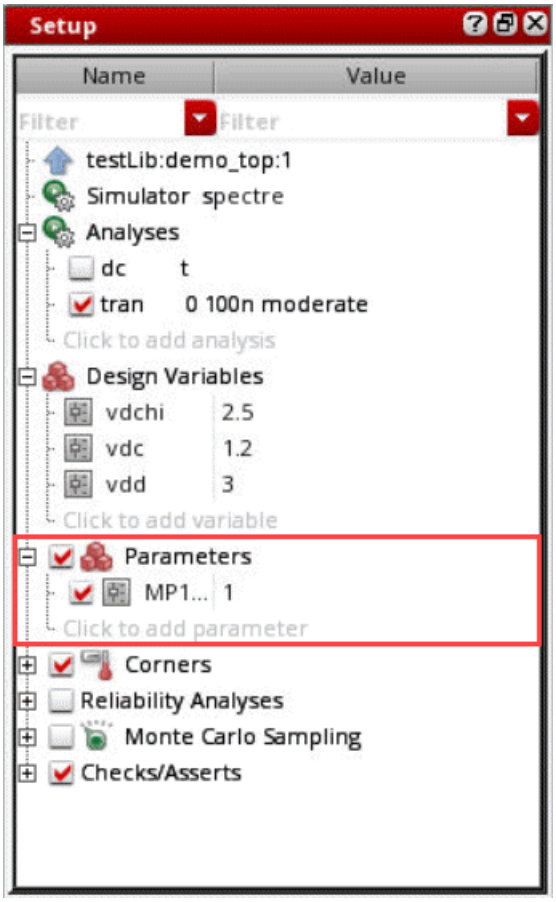
The device instance parameter name is displayed in the *Parameter* column and value is displayed in the *Value* column. You can hover above an instance name to view the related test information along with the parameter name and prompt name for that instance.




Note the following:


© 2023 Cadence Design Systems, Inc.

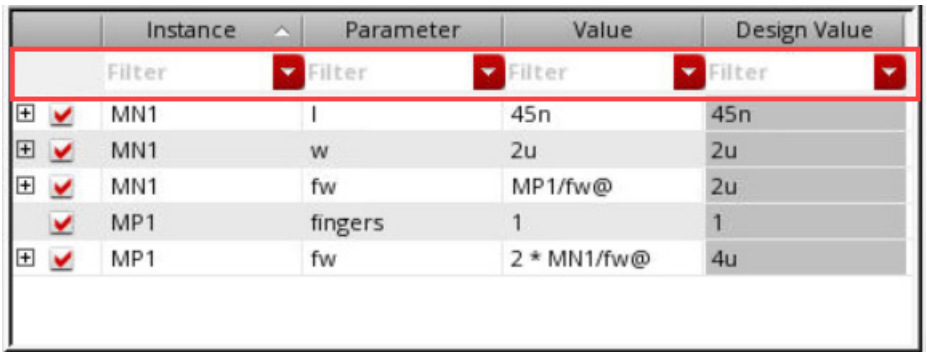
- The parameter is also displayed in the *Parameters* section in the *Setup* assistant, as shown below.



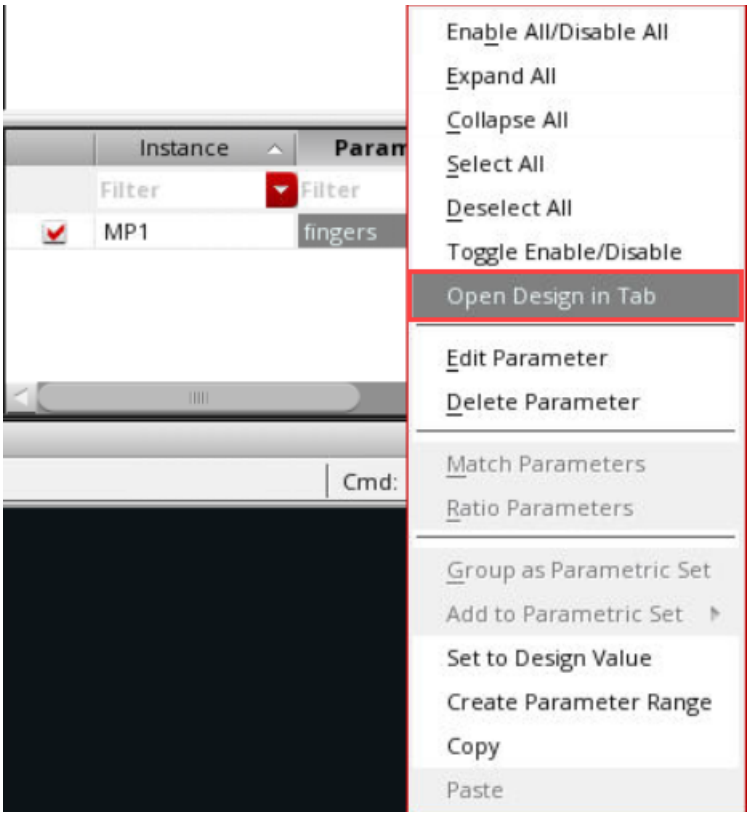

The value to be used for the parameter is same as the value of the parameter specified in the schematic, which is also shown in the *Design Value* column in the lower half of the *Parameters* tab on the *Variables and Parameters* assistant. You can specify a different value to be used while running simulation. For more details, refer to Changing the Value of Device Instance Parameters.


© 2023 Cadence Design Systems, Inc.

- If required, you can also view specific instance parameters by using the filters available in each column header of the lower half of the *Variables and Parameters* assistant.




To view the schematic where a particular parameter is set, right-click that parameter and select *Open Design in Tab*.

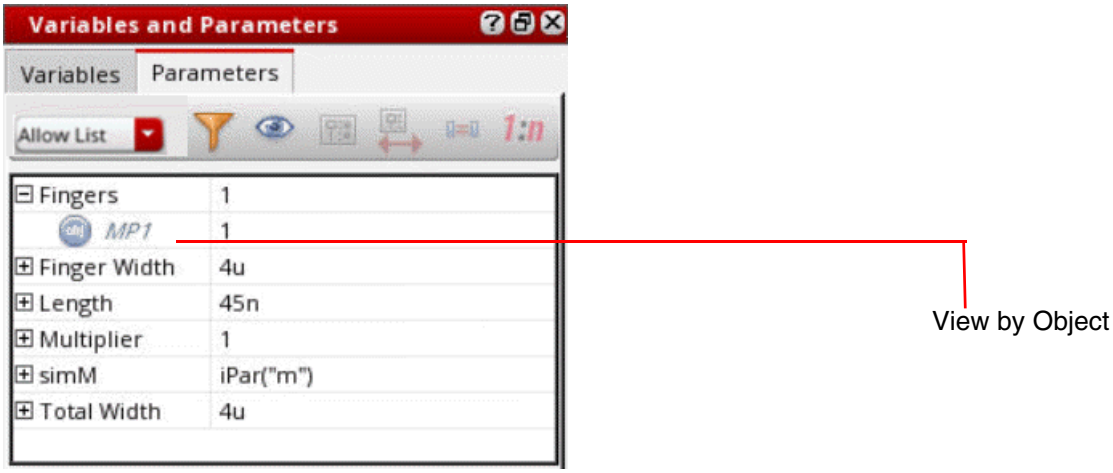



**Note:** If the schematic containing that parameter is already open, then on using this option, the existing schematic tab gets refreshed instead of opening a new one.


© 2023 Cadence Design Systems, Inc.

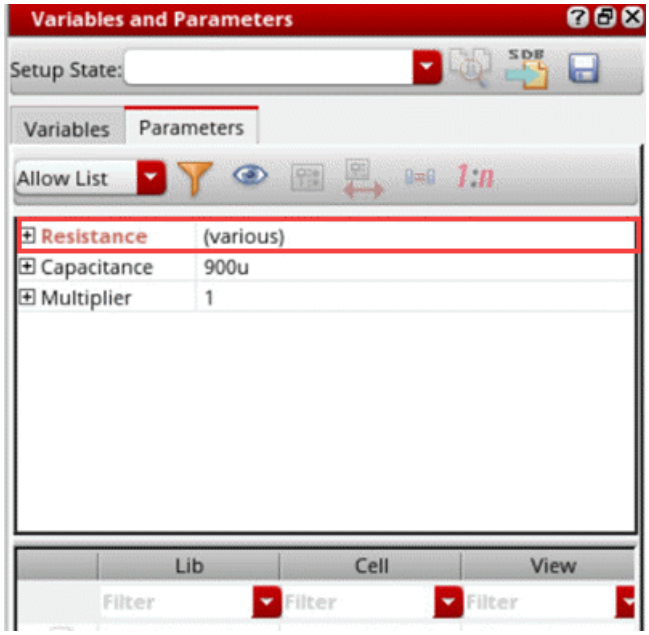
### *Additional Points to Note*

- The default view for the *Parameters* tab is `View by Property`. That is why when you select an instance in the schematic, by default, a list of property names is displayed. However, you can change the view to list the instance names by clicking the toggle on the toolbar of the *Parameters* tab, as shown below.




  You can also set the default view by setting the `defaultParametersViewBy` environment variable.

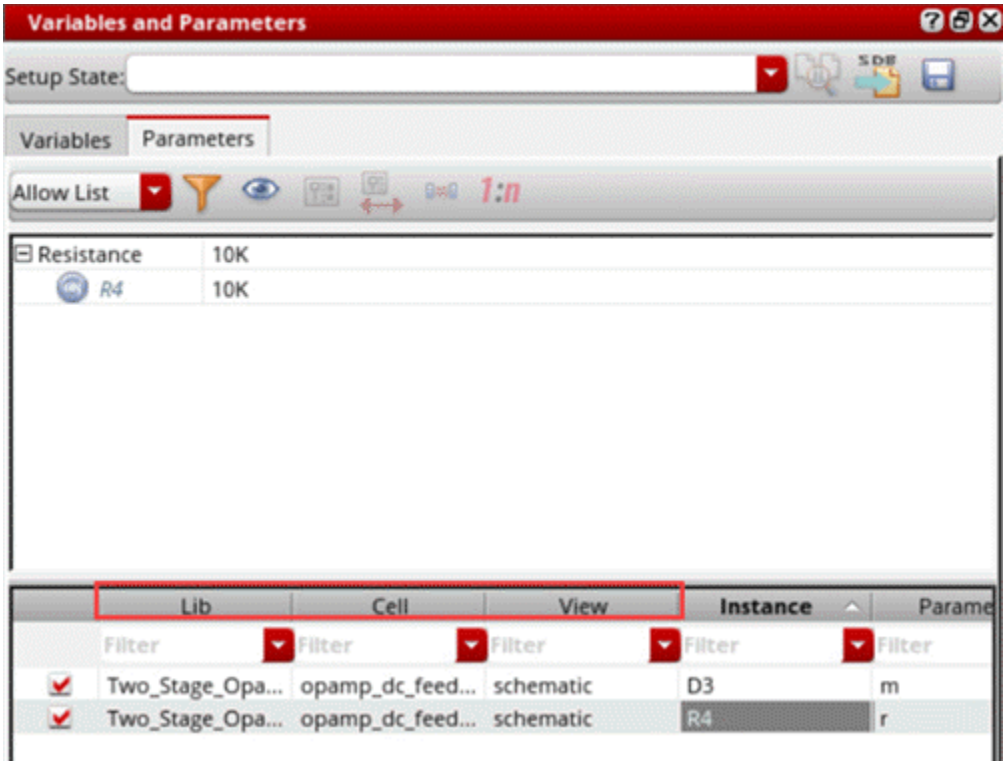
- If you have selected more than one instance in the schematic and each instance has a different design value for a property, the value is shown as `(various)`.





© 2023 Cadence Design Systems, Inc.

- When the view is set to `View by Property`, you can select multiple devices in the schematic and use the *Create Parameter* command to create parameters for all the selected devices simultaneously. For example, if you have selected two instances, `M1` and `M2`, on the schematic. In the *Variables and Parameters* assistant, right-click `Length`, and choose *Create parameter*. The `Length` parameter is created for both `M1` and `M3` simultaneously.
- You can also create matched or ratio-matched parameters by using the *Match Parameters* or *Ratio Matched Parameters* commands on the toolbar. For more details, refer to Creating Matched Device Parameters and Creating Ratio-Matched Device Parameters.
- If the devices have a large number of parameter properties, you can use filters to show only the relevant set of properties.
- When you create parameters, the lower half of the *Variables and Parameters* assistant displays the instance name, parameter name, parameter value, and the value of property in the design. You can also show the names of library, cell, and view of the instances. Right-click any column name in the lower half of the assistant and choose *Show Lib, Cell, View*. Three more columns are displayed, as shown below.




You can show or hide any one of these three columns using the commands in the context-sensitive menu for this section of the assistant. If required, you can even edit the names of library, cell, and view in this table. ADE Explorer validates the existence of the


© 2023 Cadence Design Systems, Inc.

newly specified library, cell, and view. If found, the details are updated in the assistant. Otherwise, an error is displayed in the CIW.

- When you open a saved ADE Explorer view, if the cellview details of the instance parameters are different, the *Variables and Parameters* assistant shows all the columns to display the cellview details.

# Changing the Value of Device Instance Parameters

Perform the following steps to change the value of a device instance parameter for a simulation without affecting the value on the schematic:

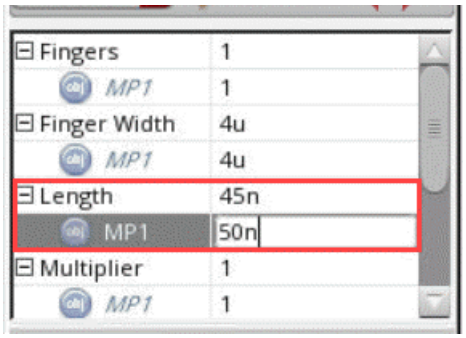
1. On the upper half of the *Parameters* tab on the *Variables and Parameters* assistant, click the value of the device instance parameter you want to change and type a new value.

   **Note:** You may have to expand the section by clicking the plus sign to the left of the instance name.

*Tip*

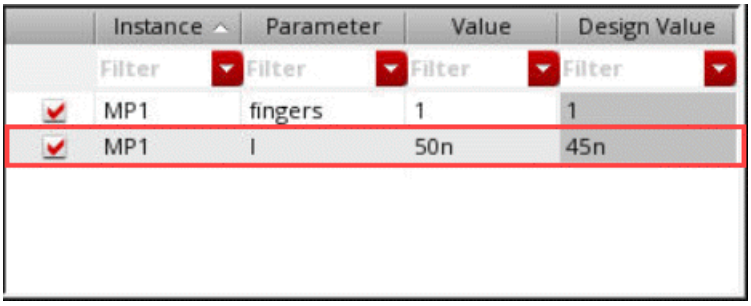
To view only the parameters in the *Parameters* tab, see Viewing Parameters by Property and Object Names.

For example, change the value of the *Length* parameter from `45n` to `50n`.




The instance parameter and its new value (`50n`) appear in the lower half of the *Parameters* tab on the *Variables and Parameters* assistant. The simulator uses this value when you click *Run Simulation*.


© 2023 Cadence Design Systems, Inc.




Alternatively, perform the following steps:

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

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

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

   The simulator uses this value when you click *Run Simulation*.

The parameter is also displayed in the *Parameters* section in the *Setup* assistant.

***Important Points to Note***

- You can add parameters only using the *Parameters* tab of the *Variables and Parameters* assistant. Parameters cannot be directly added in the *Setup* assistant.
- At any time, you can reset value of the parameter to its original value which was set in the design schematic. For this, in the lower half of the *Parameters* tab, right-click a parameter name and choose *Set to Design Value*.

# Sweeping Pcell Parameters

You can sweep Pcell parameters in the same way as described above for other device parameters. The sweep values are used to create multiple design points for simulations. During netlist creation, ADE Explorer and ADE Assembler create subcircuits for each swept Pcell parameter value.

This feature is helpful for advanced node designs that contain Pcells with parameters to adjust circuit topology changes, such as the number of stacked components represented by a device. You can run local or global optimization to identify the best value for the Pcell parameters to optimize your designs.


© 2023 Cadence Design Systems, Inc.

# Specifying Values of Device Instance Parameters

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.

Perform the following steps to specify an instance parameter as a sweep parameter:

1. Create instance parameters for the design associated with your test. See Creating Device Instance Parameters on page 496 for more information.
2. Perform one of the following steps to define sweep values for an instance parameter:
   - In the lower half of the *Parameters* tab in the *Variables and Parameters* assistant, double-click the value of the parameter and specify a sweep value.

     Alternatively, in the *Setup* assistant, double-click the value of the parameter and specify a sweep value.
   - In the lower half of the *Parameters* tab in the *Variables and Parameters* assistant, double-click the value of the parameter, then click the ellipses button to open the *Parameterize* form and define a parameter specification.

     Alternatively, in the *Setup* assistant, double-click the value of the parameter, then click the ellipses button to open the *Parameterize* form and define a parameter specification.

     See Adding or Modifying Specifications for Parameters and Design Variables on page 520 for more information.

To specify a valid sweep value, ensure that:

- The value is not a blank string or does not contain a whitespace
- If contains a range, it has the start, step, and stop values separated by colons, commas, spaces, or a combination of these
- If specified as an expression, it is written in correct SKILL syntax and contains balanced parenthesis and quotes
- If contains a file path, it is enclosed in double quotes so that it is not evaluated

# Creating Matched Device Parameters

You can create matched parameters for device matching such that one device's parameters track with another's.


© 2023 Cadence Design Systems, Inc.

Perform the following steps after creating device instance parameters, to specify two or more devices to match:

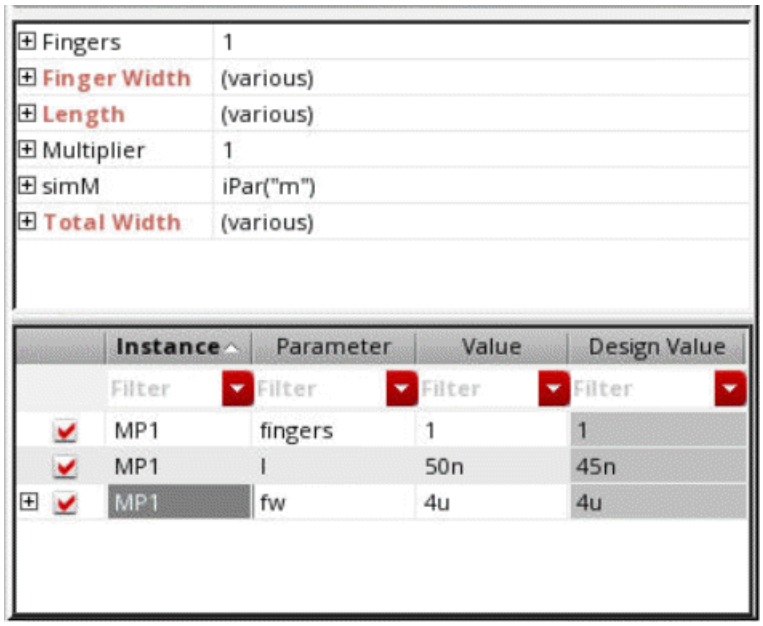
1. On the schematic, click to add the first device, then *Shift*+click to add one or more additional devices.

   As you click, each device and its parameters appear on the upper half of the *Parameters* tab of the *Variables and Parameters* assistant.

2. On the upper half of the *Parameters* tab of the *Variables and Parameters* assistant, select (highlight) the device you want as the primary device, against which all other devices are matched.

3. Click *Match Parameters* on the toolbar of the *Variables and Parameters* assistant.

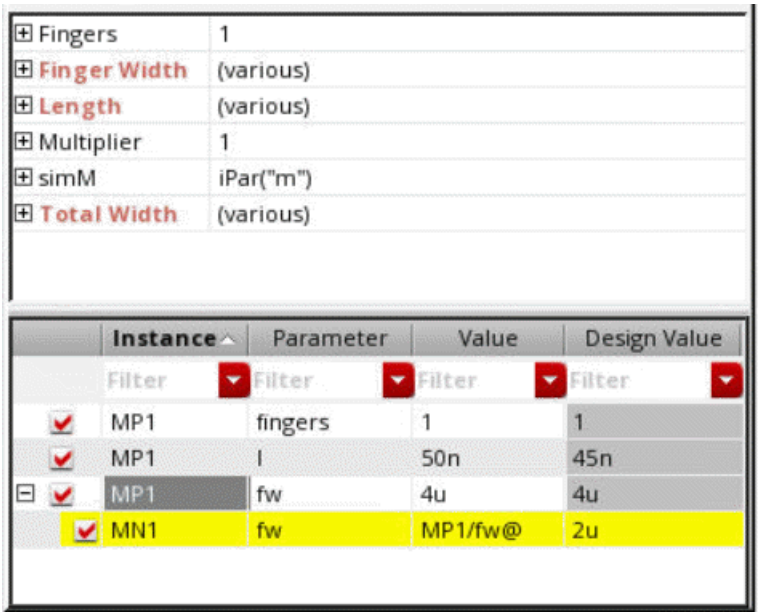
   The instance name of the primary device for which matched parameters exist appears in the lower half of the *Parameters* tab, in the *Variables and Parameters* assistant.





© 2023 Cadence Design Systems, Inc.

4. Optionally, to view the devices whose parameters are matched to the primary device, click the plus sign to the left of an instance name in the lower half of the *Parameters* tab.




The value of a matched parameter appears in the *Value* column as shown below, to indicate the matching relationship:

*primaryInstName/deviceParam@*

For example, in the above figure, the value `MP1/fw@` for instance `MN1` indicates that the `fw` parameter of instance `MP1` is matched to the `fw` parameter of instance `MN1` (the primary device).

**Note:** You can also match parameters across different levels of design hierarchy.

*Tip*

Right-click 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.

# 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.

Perform the following steps to specify two or more devices to ratio-match:


© 2023 Cadence Design Systems, Inc.

1. On the schematic, click to add the first device. Next, *Shift*+click to add each additional device.

   As you click, each device and its parameters appear in the panel on the upper half of the *Parameters* tab of the *Variables and Parameters* assistant.

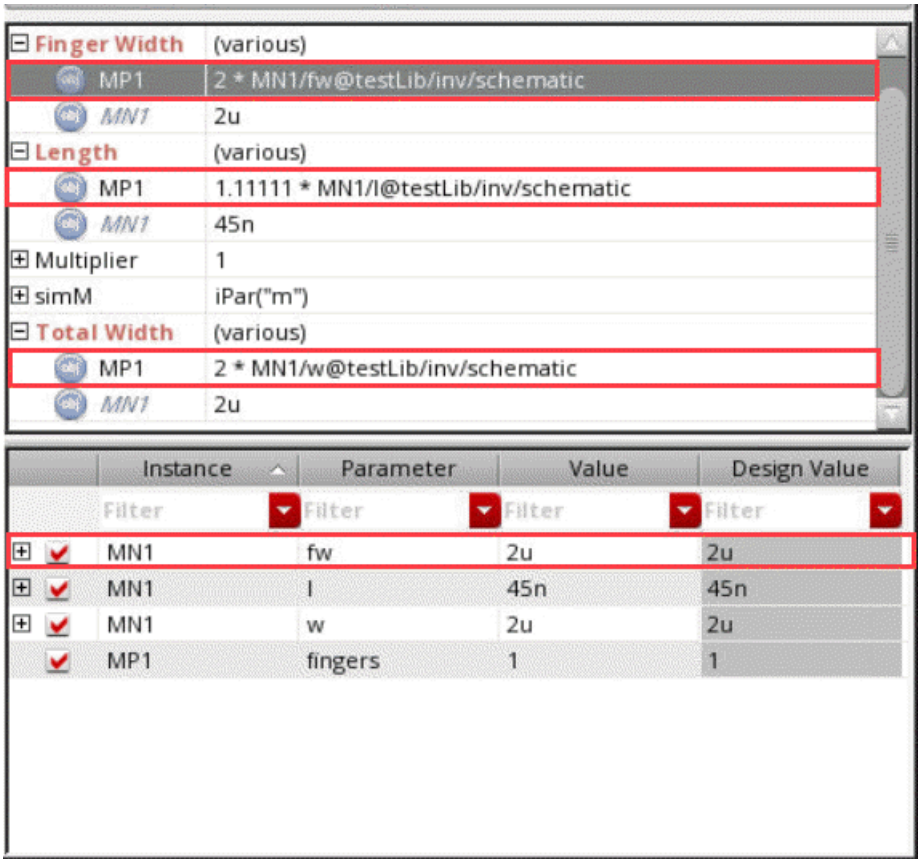
   **Note:** You can also create ratio-matched parameters for the devices in different hierarchy levels of the design.

2. Select a device in the upper section on the *Parameters* tab of the *Variables and Parameters* assistant.

3. Click *Ratio Matched Parameters* 1:n.

   The ratio-matched parameters are created. A row is created in the lower section of the assistant and name of the primary device (that appeared on top in the upper section of this assistant) and the parameter name is displayed in that.

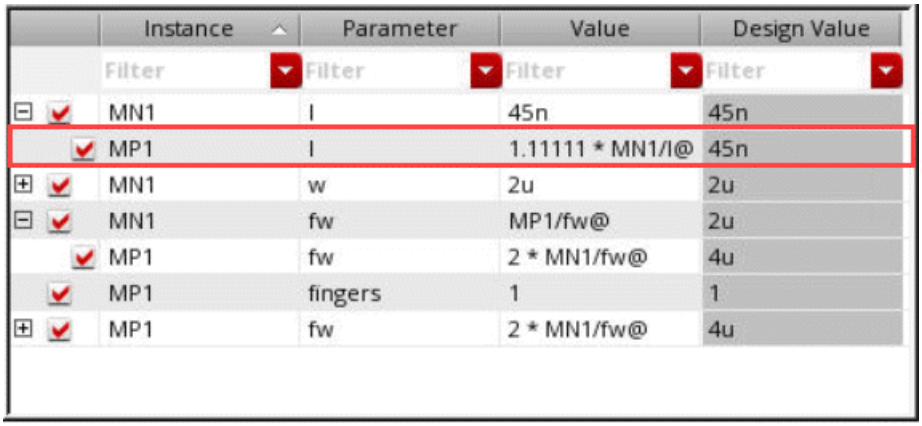
   For example, if you select two devices, `MP1`, and `MN1`, and create a ratio-matched parameter for the parameter `length`, finger width and total width a new row displaying `MN1` and `fw` is created, as shown below.





© 2023 Cadence Design Systems, Inc.

4. Optionally, to view the devices whose parameters are ratio-matched to the primary device, click the plus (+) sign to the left of an instance name in the lower half of the *Parameters* tab.




The value of a ratio-matched parameter appears in the *Value* column, as shown below, to indicate the matching relationship:

*ratio*primaryInstName/deviceParam@*

For example, in the above figure, the value 1.25*`M1/l@` for instance `M2` indicates that the `l` parameter of instance `M2` is ratio-matched to the `l` parameter of instance `M1` (the primary device).

*Tip*

You can also right-click 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.

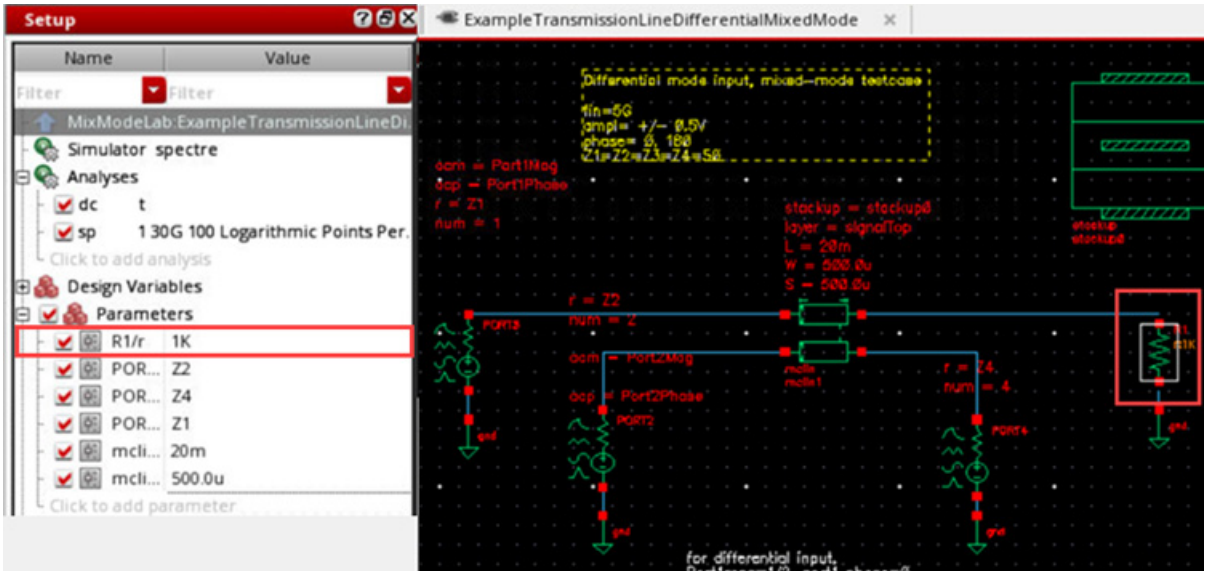
# Deleting a Device Parameter

Perform the following step to delete a parameter:

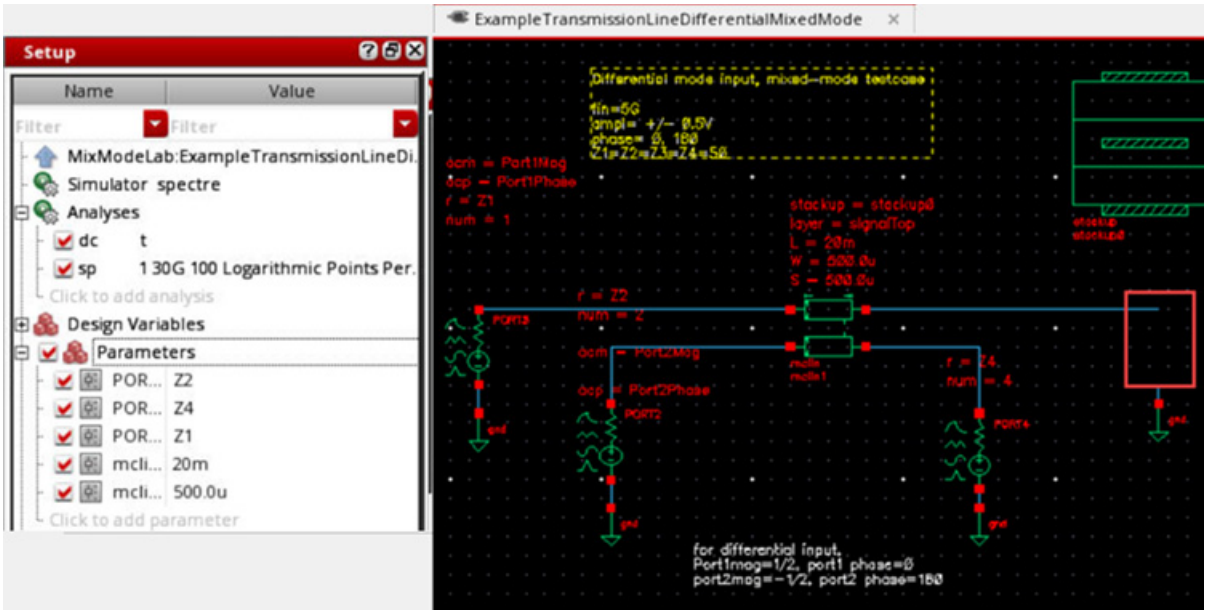
- In an expanded *Parameters* section on the *Setup* assistant or on the lower part of the *Parameters* tab of the *Variables and Parameters* assistant, right-click the parameter you want to delete and choose *Delete Parameter*.


© 2023 Cadence Design Systems, Inc.

In case you wish to delete the parameters that are not being used in the design anymore, right-click the *Parameters* section in the Setup assistant and choose *Delete Unused Parameters* from the context menu. For example, consider the following setup.




On choosing the *Delete Unused Parameters* command, the parameter `PORT3/r` gets deleted since the resistance `Z3` is does not exist in the design anymore.





© 2023 Cadence Design Systems, Inc.

# 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.

Perform the following step to enable/disable a parameter:

➥ In the *Parameters* section on the *Setup* assistant or the *Parameters* tab on the *Variables and Parameters* assistant, select/deselect the check box to the left of the parameter.

Alternatively, right-click the parameter and choose *Enable/Disable* from the context-sensitive menu.

**Note:** When a parameter is disabled, its value given on the *Setup* assistant is not used for simulation, but its definition is retained in the *Parameters* tree on the *Setup* assistant and the *Parameters* tab on the *Variables and Parameters* assistant. 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 context-sensitive menu.

Perform the following step to enable/disable all parameters:

➥ On the *Setup* assistant, select/deselect the check box to the left of *Parameters*.

Alternatively, right-click the *Parameters* tab of the *Variables and Parameters* assistant and choose *Enable All/Disable All*.

Perform the following step to toggle the enabling or disabling of one or more parameters:

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

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

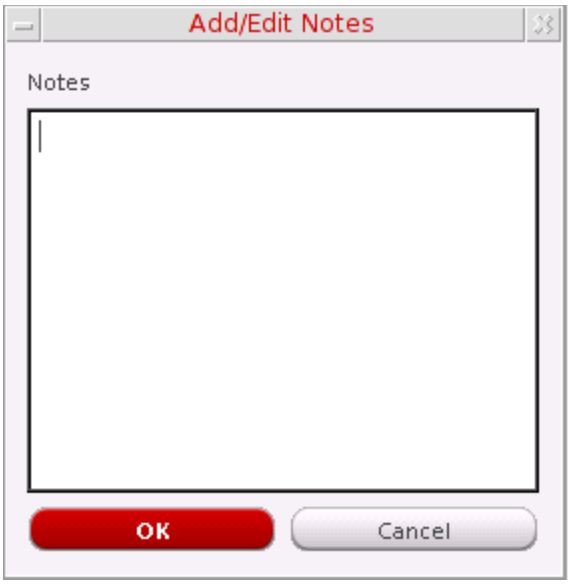
# Adding Notes to Parameters

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

1. In the *Setup* assistant, expand the *Parameters* section, right-click the parameter for which you want to add notes, and choose *Notes*.


© 2023 Cadence Design Systems, Inc.

The *Add/Edit Notes* form is displayed.

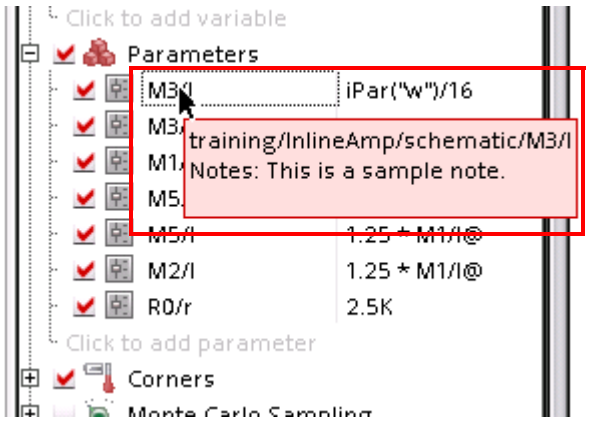



2. In the *Notes* field, add notes for the device parameter.

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

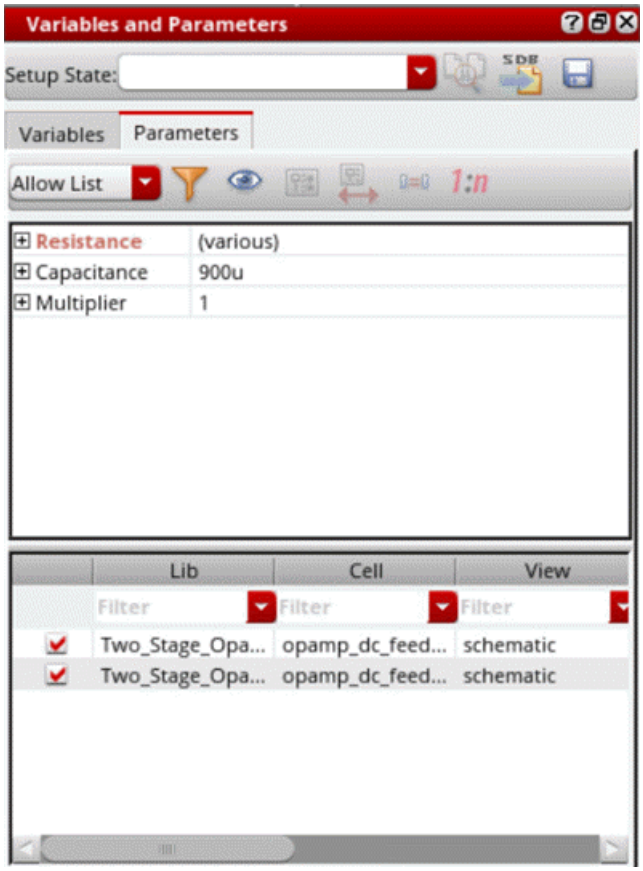




© 2023 Cadence Design Systems, Inc.

# Viewing Parameters by Property and Object Names

By default, in the upper half of the *Parameters* tab in the *Variables and Parameters* assistant, the parameters are displayed by property names, as shown below.



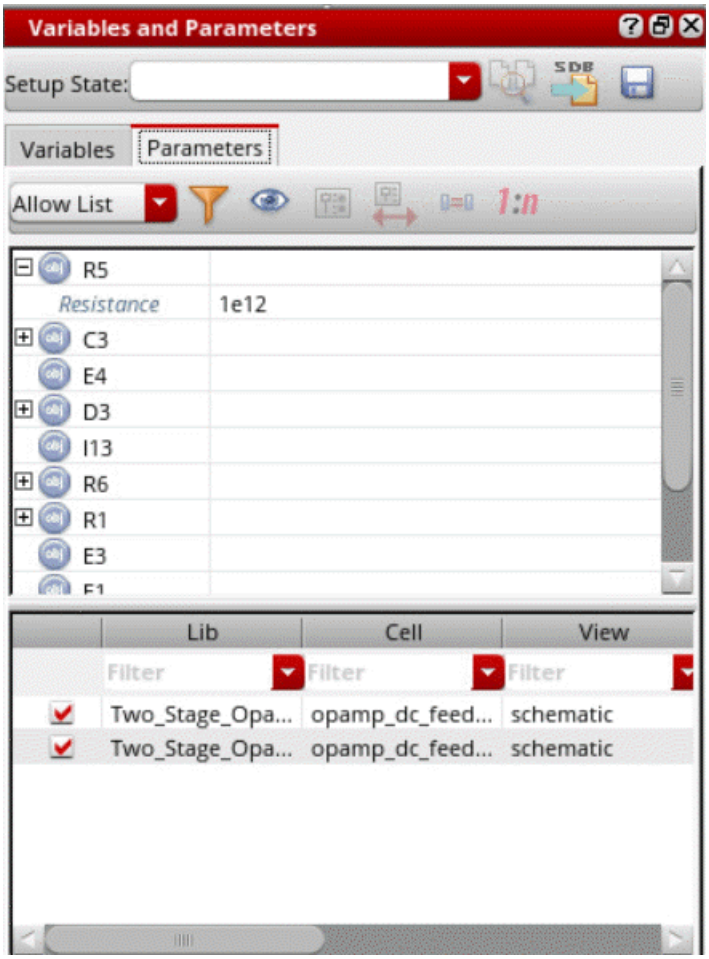

Perform the following step to view the parameters by object names:

➡ Right-click the upper part of the *Parameters* tab and choose *View by Object*.

Alternatively, click ( ) to toggle the view types.


© 2023 Cadence Design Systems, Inc.

The view changes to a section of objects.




**Note:** You can set the preferred view as the default view by using the `defaultParametersViewBy` environment variable.

# Filtering Device Instance Parameters

Perform the followings step to filter the device instance parameters that appear on the upper half of the *Parameters* tab on the *Variables and Parameters* assistant:

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

| | |
|---|---|
| `Allow List` | Displays parameters specified in the list that you can edit by using the Filter Setup form. |

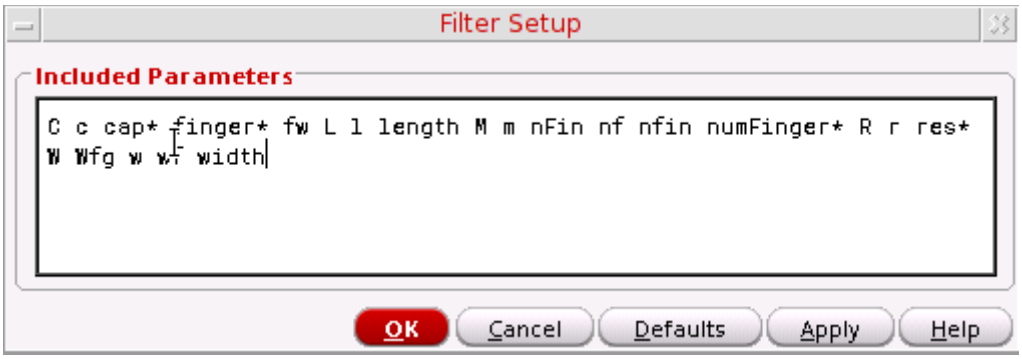

© 2023 Cadence Design Systems, Inc.

| | |
|---|---|
| `Automatic` | Displays all editable CDF parameters with non-nil CDF values. |
| `Editable` | Displays all editable parameters. |
| `All` | Displays all the parameters. |

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

By default, the Allow List filter shows the parameters set by the `defaultParametersAllowList` environment variable. You can edit this list in either of the following ways:

- By editing the `defaultParametersAllowList` environment variable.
- By using the *Filter Setup* form which is displayed when you click *Allow List Filter* ( ) on the toolbar.




You can use the `*` wildcard character to specify the parameter names in the Allow List filter. For example, `*inger* res*`.

The *Allow List Filter* command is enabled only when the *Filter* drop-down list is set to `Allow List`. For other filter types, this is not applicable.

**Note:** If you have defined any custom device filter using the `axlRegisterCustomDeviceFilter` function, the name of that filer also appears in the drop-down list. See Creating Custom Device Filters for more information. However, it is recommended to use the `Allow List` filter to customize the list of parameters to be displayed.


© 2023 Cadence Design Systems, Inc.

# Creating Custom Device Filters

Perform the following steps to create a custom device instance parameter filter:

1. Write a procedure to define your custom filter.

   For example, you want to filter out the `Total Width`, `Threshold` and `S/D Metal Width` parameters for all instances of the `pmos2v` cell, write a procedure as shown below:

```
(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.


© 2023 Cadence Design Systems, Inc.

*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 shown below:

   ```
   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 for the selected device instance(s).

# Changing the Order of Device Parameters

When there are multiple parameters in the *Parameters* list of the *Setup* assistant, you can change the order in which they are listed. A change in the order of device parameters in this assistant changes the nesting of sweep points as well.

Perform the following steps in the *Setup* assistant to change the order of a parameter:

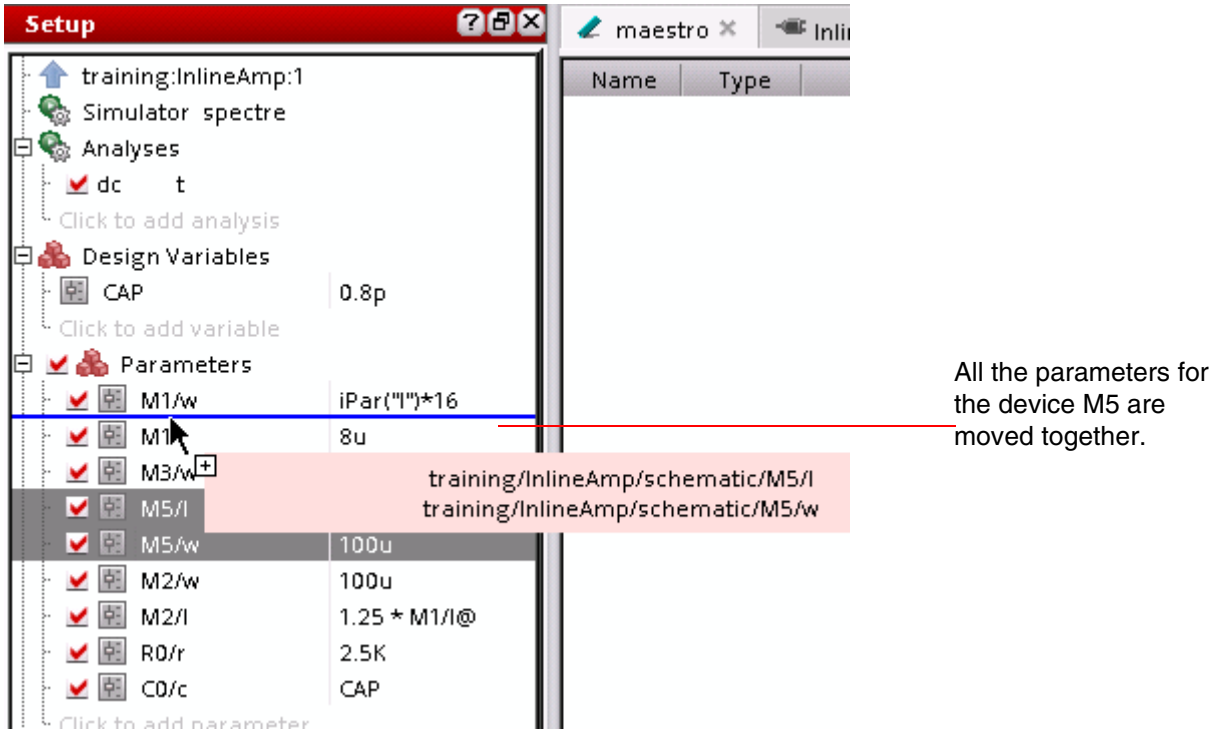
1. Expand the *Parameters* section to view all the parameters.
2. Click a parameter name and drag it.

   A blue indicator line appears at the location where you have moved it. The tooltip also shows the name of the parameter being moved.

   Also, because all the parameters for a device are placed together, you cannot move only a single parameter for a device to another location in the list. Therefore, while you are


© 2023 Cadence Design Systems, Inc.

moving a device parameter, all the other parameters for the device are also highlighted and moved, as shown below.




**Note:** While you are dragging the parameters, you cannot place the device parameters in between the list of parameters for another device. If you drop the parameters in the midst of the list of parameters for another device, they are placed above the list if you are moving the parameters up, or below the list if you are moving the parameters down.

3. Drag the indicator line to the desired location and drop the parameter.

You can also select the parameters for more than one device and move them together.

# Disabling Callbacks on Swept Device Parameters

By default, the program executes callbacks on device parameters that you sweep.

Perform the following step to disable callbacks on swept device parameters:

➡ In your `.cdsinit` file, set the `axlExecuteCallbacks` variable to `nil`:

```
axlExecuteCallbacks=nil
```


© 2023 Cadence Design Systems, Inc.

For more information about callbacks in parameter expressions, see "*Triggering callbacks*" in the "*Advice and Warnings*" appendix of the *Component Description Format User Guide*.

# Adding or Modifying Specifications for Parameters and Design Variables

Perform the following steps to add or change the specification for a parameter or a design variable you want to sweep (vary):

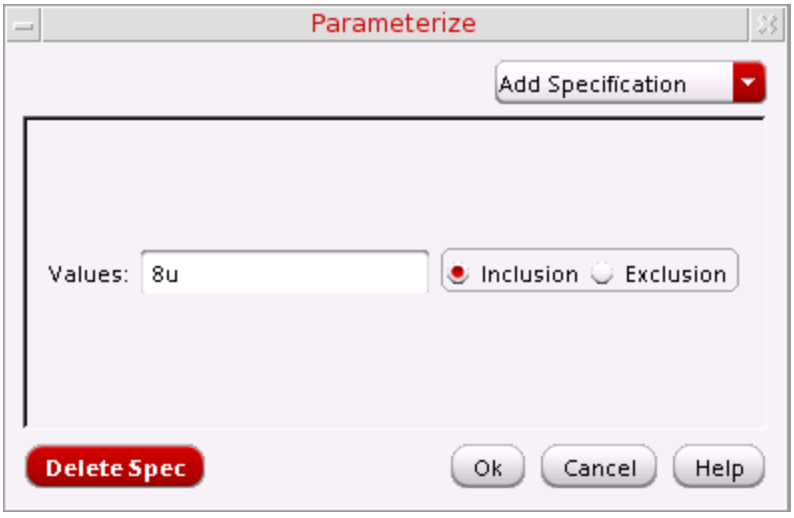
1. Perform the following steps to open the *Parameterize* form and define a specification for parameters or design variables.

   - To add or change specifications for a parameter, right-click the parameter in the *Parameters* section of the Setup assistant or in the *Parameters* tab of the Variables and Parameters assistant and choose *Edit Parameter*.

     Alternatively, double-click the *Value* field next to the parameter in the Setup assistant or in the *Parameters* tab of the Variables and Parameters assistant, then click the ellipses button.

   - To add or change specifications for a design variable, double-click the *Value* field next to the variable in the Setup assistant or in *Variables* tab of the Variables and Parameters assistant, then click the ellipses button.

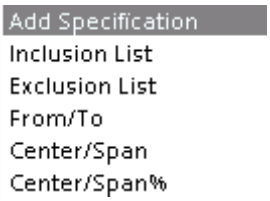
   The *Parameterize* form appears.





© 2023 Cadence Design Systems, Inc.

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 parameter.
- 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 parameter and a method for stepping through that range.
- Center/Span values — You can specify a center value and a span value for varying your parameter.
- Center/Span% values — You can specify a center value and a percentage span value for varying your parameter.

## Specifying an Inclusion List of Values

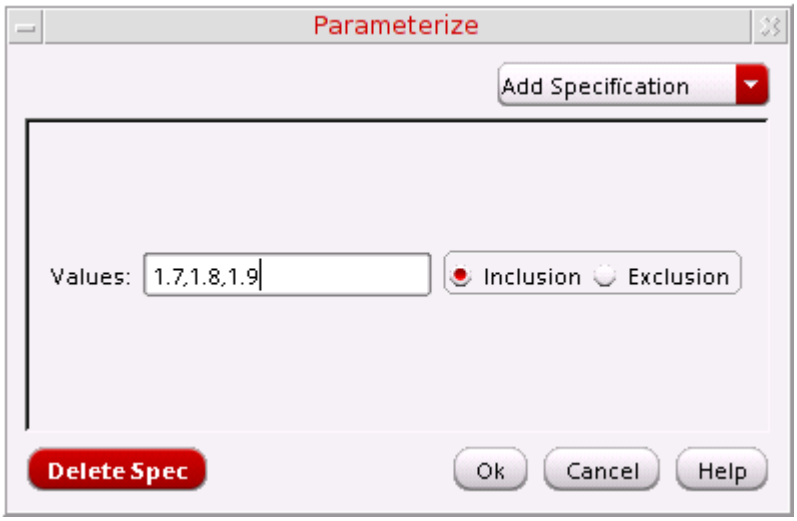
Perform the following steps to specify an inclusion list of values:

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 selected.


© 2023 Cadence Design Systems, Inc.

2. In the *Values* field, type a list of values you want to include in your sweep. You can separate the values in the list with either a comma or a space.




3. Optionally, 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 522.

## Specifying an Exclusion List of Values

One application of the exclusion list is to define a *From/To* parameter specification (see Specifying a Range of Values on page 523) and exclude some values from the range by defining an exclusion list.

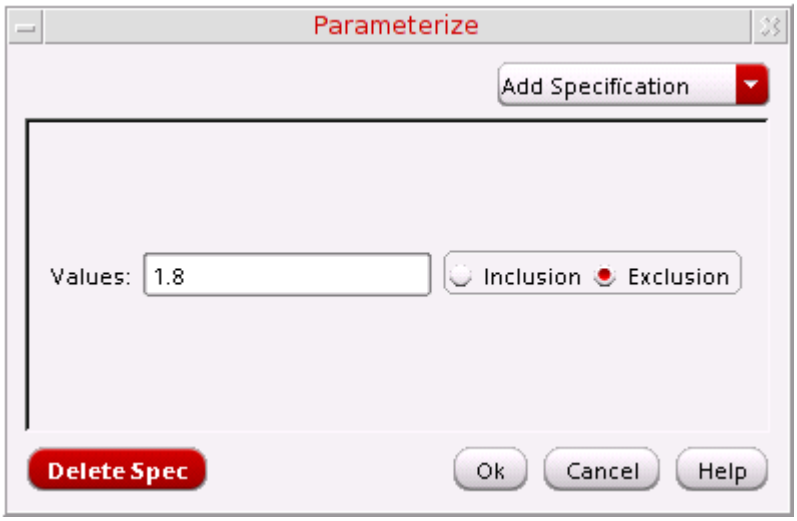
Perform the following steps to specify an exclusion list of values:

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 selected.


© 2023 Cadence Design Systems, Inc.

2. In the *Values* field, type a list of values you want to exclude from your sweep. You can separate the values in the list with either a comma or a space.




3. Optionally, 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 521.

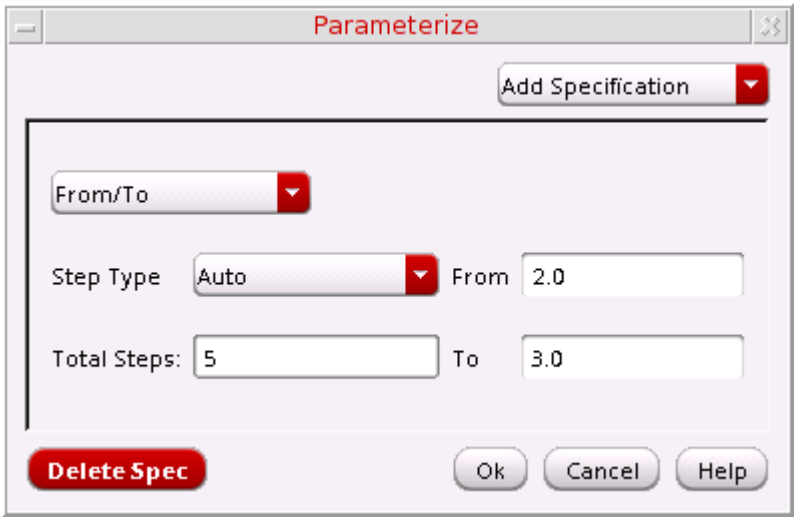
## Specifying a Range of Values

Perform the following steps to specify a range of values and sweep through those values:

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


© 2023 Cadence Design Systems, Inc.

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 |
| *LinearStepCount* | Linear steps from the *From* value to the *To* value |


© 2023 Cadence Design Systems, Inc.

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<br>**Note:** Typing `10` for *Multiplier* is tantamount to selecting *Decade* and specifying `1` step per decade. |
| *LinearStepCount* | *Step Count* | Number of steps to take |

6. Optionally, 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 525 and Specifying Center and Span as a Percentage on page 527 for more information.

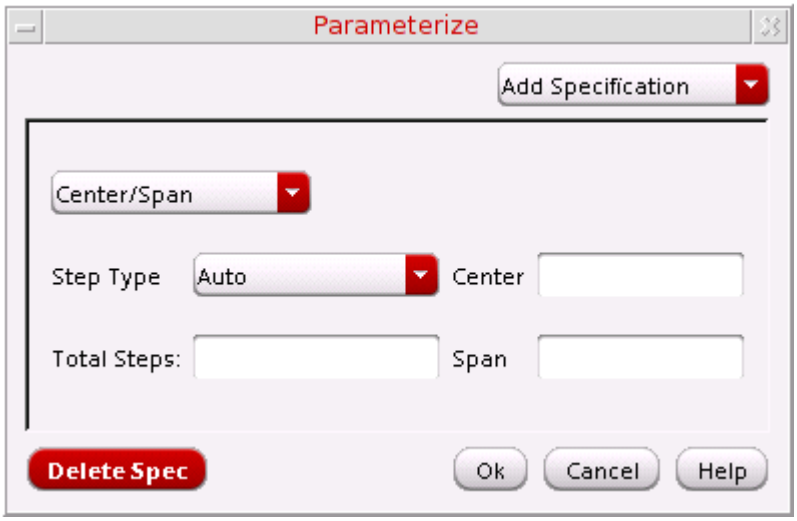
## Specifying Center and Span

Perform the following steps to specify a center value and a span value:

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


© 2023 Cadence Design Systems, Inc.

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/2* to *Center+Span/2* |
| *Decade* | Decade steps from *Center−Span/2* to *Center+Span/2* |
| *Octave* | Octave steps from *Center−Span/2* to *Center+Span/2* |
| *Logarithmic* | Logarithmic steps from *Center−Span/2* to *Center+Span/2* |
| *Times* | Steps taken according to a specified multiplier from *Center−Span/2* to *Center+Span/2* |
| *LinearStepCount* | Linear steps from *Center−Span/2* to *Center+Span/2* |


© 2023 Cadence Design Systems, Inc.

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<br>**Note:** Typing `10` for *Multiplier* is tantamount to selecting *Decade* and specifying `1` step per decade. |
| *LinearStepCount* | *Step Count* | Number of steps to take |

6. Optionally, 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 527 and Specifying a Range of Values on page 523 for more information.

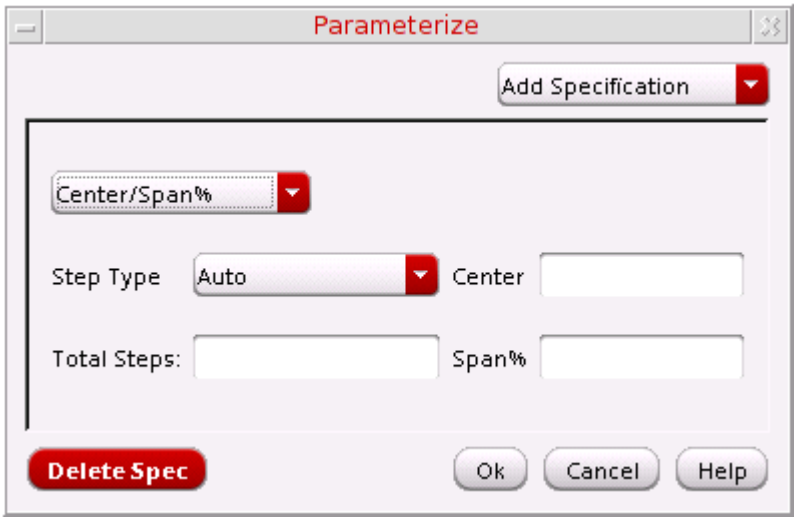
## Specifying Center and Span as a Percentage

Perform the following steps to specify a center value and a span percentage:

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


© 2023 Cadence Design Systems, Inc.

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 $Center - \frac{(Span\% \times Center)}{100}/2$ and $Center + \frac{(Span\% \times Center)}{100}/2$ 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 $Center - \left(\frac{(Span\% \times Center)}{100}\right)/2$ to $Center + \left(\frac{(Span\% \times Center)}{100}\right)/2$ |
| *Decade* | Decade steps from $Center - \left(\frac{(Span\% \times Center)}{100}\right)/2$ to $Center + \left(\frac{(Span\% \times Center)}{100}\right)/2$ |


© 2023 Cadence Design Systems, Inc.

*Octave* Octave steps from

$$Center - \left(\frac{(Span\% \times Center)}{100}\right)/2 \quad \text{to}$$

$$Center + \left(\frac{(Span\% \times Center)}{100}\right)/2$$

*Logarithmic* Logarithmic steps from

$$Center - \left(\frac{(Span\% \times Center)}{100}\right)/2 \quad \text{to}$$

$$Center + \left(\frac{(Span\% \times Center)}{100}\right)/2$$

*Times* Steps taken according to a specified multiplier from

$$Center - \left(\frac{(Span\% \times Center)}{100}\right)/2 \quad \text{to}$$

$$Center + \left(\frac{(Span\% \times Center)}{100}\right)/2$$

*LinearStepCount* Linear steps from

$$Center - \left(\frac{(Span\% \times Center)}{100}\right)/2 \quad \text{to}$$

$$Center + \left(\frac{(Span\% \times Center)}{100}\right)/2$$

**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<br>**Note:** Typing `10` for *Multiplier* is tantamount to selecting *Decade* and specifying `1` step per decade. |
| *LinearStepCount* | *Step Count* | Number of steps to take |


© 2023 Cadence Design Systems, Inc.

6. Optionally, 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 525 and "Specifying a Range of Values" on page 523 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 context-sensitive menu.
(*Delete Specification* is the only item on this context-sensitive 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*.

   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*.

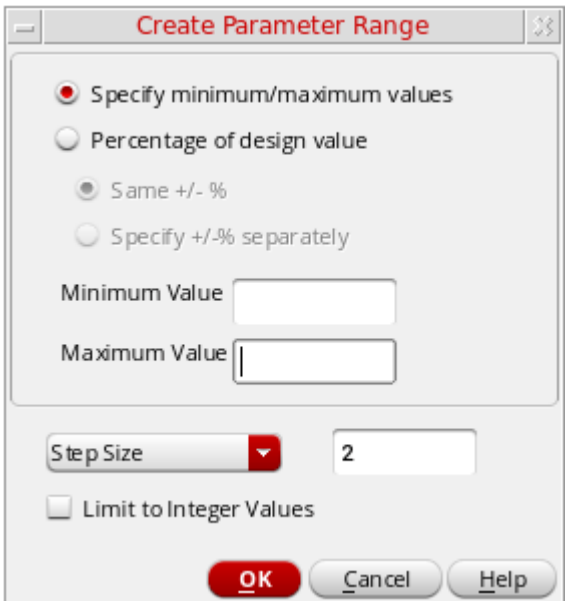

© 2023 Cadence Design Systems, Inc.

# Creating Parameter Ranges

To specify the range for an already existing parameter,

- Open the Variables and Parameters assistant and edit the parameter value manually, as shown in the figure below.

- Double-click in the parameter value row in the Setup pane and edit the parameter value manually.
- In the Variables and Parameters assistant, right-click one or more variables in the Parameters list for which you want to create parameter range and choose *Create Parameter Range*.
- Select the parameters and click the button.
- Right-click the parameter and choose *Create Parameter Range*. The Create Parameter Range form appears.




- 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

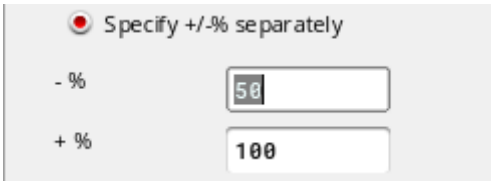

© 2023 Cadence Design Systems, Inc.

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.



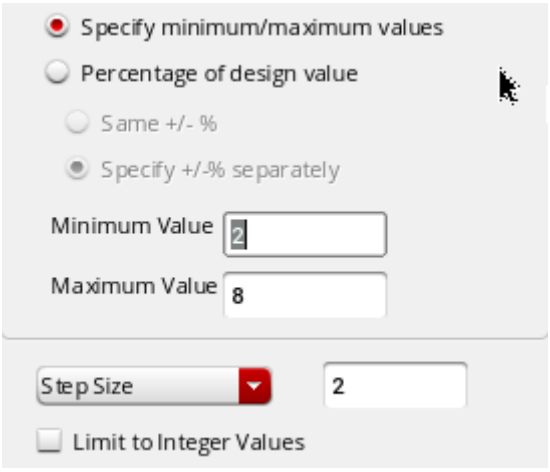

In these fields, you can specify separate % values for the upper and lower range limits.

**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


© 2023 Cadence Design Systems, Inc.

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 the parameter name and choose *Set to Design Value*.

- 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.
- 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.


© 2023 Cadence Design Systems, Inc.

- Click *OK*

  ADE Explorer creates the parameter ranges for all the parameters and updates the Setup 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 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.


© 2023 Cadence Design Systems, Inc.

# Working with Parametric Sets

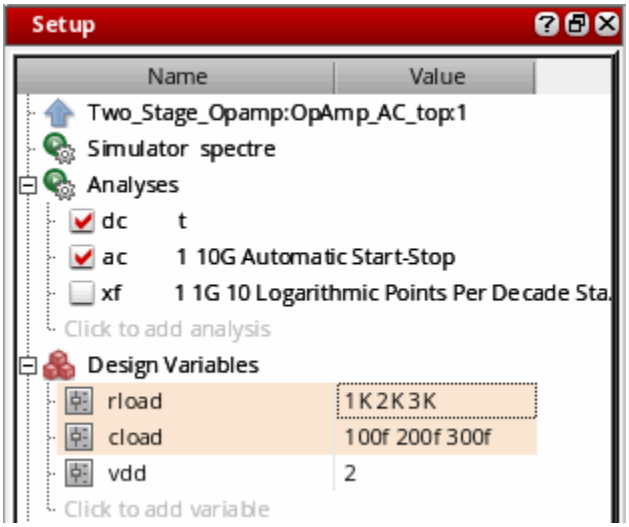
By default, ADE Explorer creates all the possible sweep combinations by pairing each value of a variable or parameter with all given values of the other variables or parameters. In parametric sets, a selected set of sweep combinations are created by picking values from the same ordinal position for all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

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.

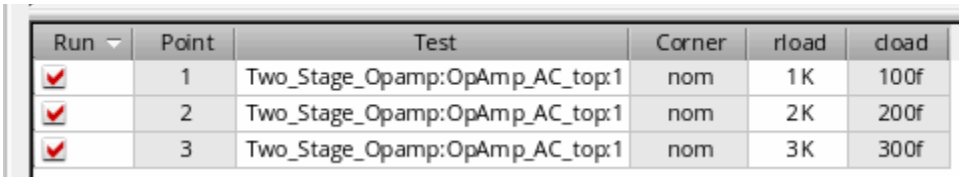
Video

You can also view the Parametric Sets in Design Variable video is available on Cadence Online Support.

Consider an example where two variables, `cload` and `rload`, are grouped together to create a parametric set, as shown in the figure below:




To view the sweep value sets that are run for this setup, open the Run Preview tab. The following three sweep value sets are created and displayed in the Points table.



| Run | Point | Test | Corner | rload | cload |
|---|---|---|---|---|---|
| ☑ | 1 | Two_Stage_Opamp:OpAmp_AC_top:1 | nom | 1K | 100f |
| ☑ | 2 | Two_Stage_Opamp:OpAmp_AC_top:1 | nom | 2K | 200f |
| ☑ | 3 | Two_Stage_Opamp:OpAmp_AC_top:1 | nom | 3K | 300f |


© 2023 Cadence Design Systems, Inc.

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 Explorer created three design points, as shown in the figure above.

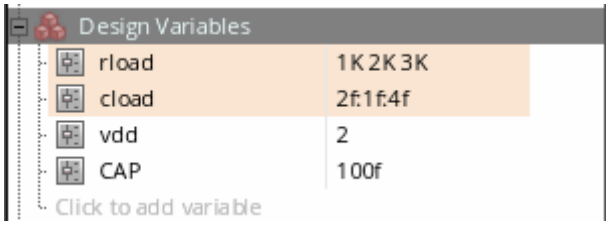
As compared to this, if the parametric sets are not created, ADE Explorer creates nine design points, where each value of `rload` is paired with all the three values of `cload`.

| Run | Point | Test | Corner | rload | cload |
|---|---|---|---|---|---|
| ☑ | 1 | Two_Stage_Opamp:OpAmp_AC_top:1 | nom | 1K | 100f |
| ☑ | 2 | Two_Stage_Opamp:OpAmp_AC_top:1 | nom | 1K | 200f |
| ☑ | 3 | Two_Stage_Opamp:OpAmp_AC_top:1 | nom | 1K | 300f |
| ☑ | 4 | Two_Stage_Opamp:OpAmp_AC_top:1 | nom | 2K | 100f |
| ☑ | 5 | Two_Stage_Opamp:OpAmp_AC_top:1 | nom | 2K | 200f |
| ☑ | 6 | Two_Stage_Opamp:OpAmp_AC_top:1 | nom | 2K | 300f |
| ☑ | 7 | Two_Stage_Opamp:OpAmp_AC_top:1 | nom | 3K | 100f |
| ☑ | 8 | Two_Stage_Opamp:OpAmp_AC_top:1 | nom | 3K | 200f |
| ☑ | 9 | Two_Stage_Opamp:OpAmp_AC_top:1 | nom | 3K | 300f |

### Creating Parametric Sets

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

1. Press 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 the sweep values for the variables. For different variables, you can specify sweep values in different formats. For example, you can specify values for variables, as shown below.




If you want to specify a sweep, the minimum number of values required for variable is `2`. If the number of sweep points for variables in the parametric set are not the same, the number of design points will be run based on the minimum points specified in the variable value. In addition, when only one value is specified for a variable in the parametric set, then that variable is considered as a scalar. For example, consider you have the following variables specified in the parametric set:


© 2023 Cadence Design Systems, Inc.

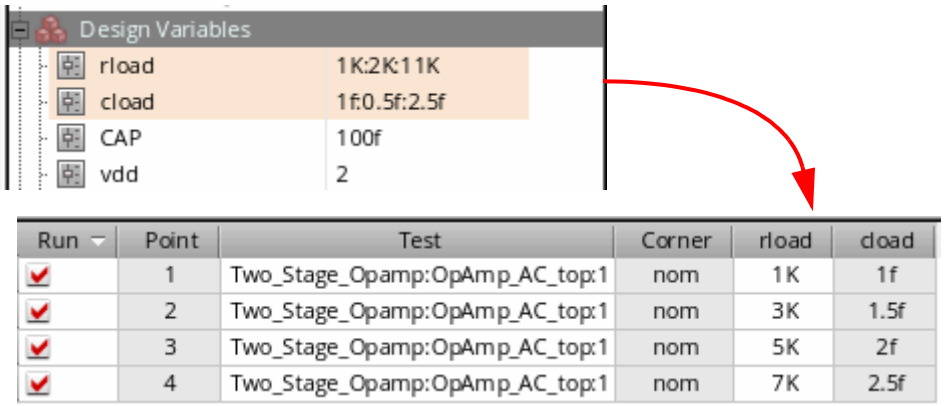
```
A=10 20 30

B=10 20

C=10
```

In this case, the following two simulation points will be run:

```
A=10 B=10 C=10

A=20 B=20 C=10
```

The variable `C` is added in both the simulation points.

Consider another example in which you have two variables, `rload` and `cload`, in the parametric set with `6` and `4` values, respectively. In this case, the minimum number of points of both the variables will be run, which is `4`.




| Run | Point | Test | Corner | rload | cload |
|---|---|---|---|---|---|
| ☑ | 1 | Two_Stage_Opamp:OpAmp_AC_top:1 | nom | 1K | 1f |
| ☑ | 2 | Two_Stage_Opamp:OpAmp_AC_top:1 | nom | 3K | 1.5f |
| ☑ | 3 | Two_Stage_Opamp:OpAmp_AC_top:1 | nom | 5K | 2f |
| ☑ | 4 | Two_Stage_Opamp:OpAmp_AC_top:1 | nom | 7K | 2.5f |

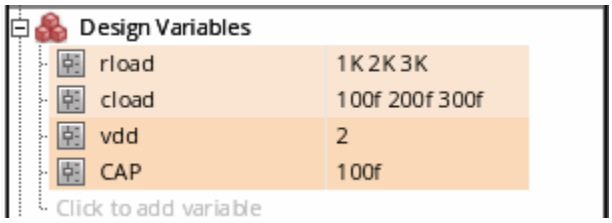
### Important Points to Note

Note the following changes in the Setup assistant and the Variables and Parameters assistant:

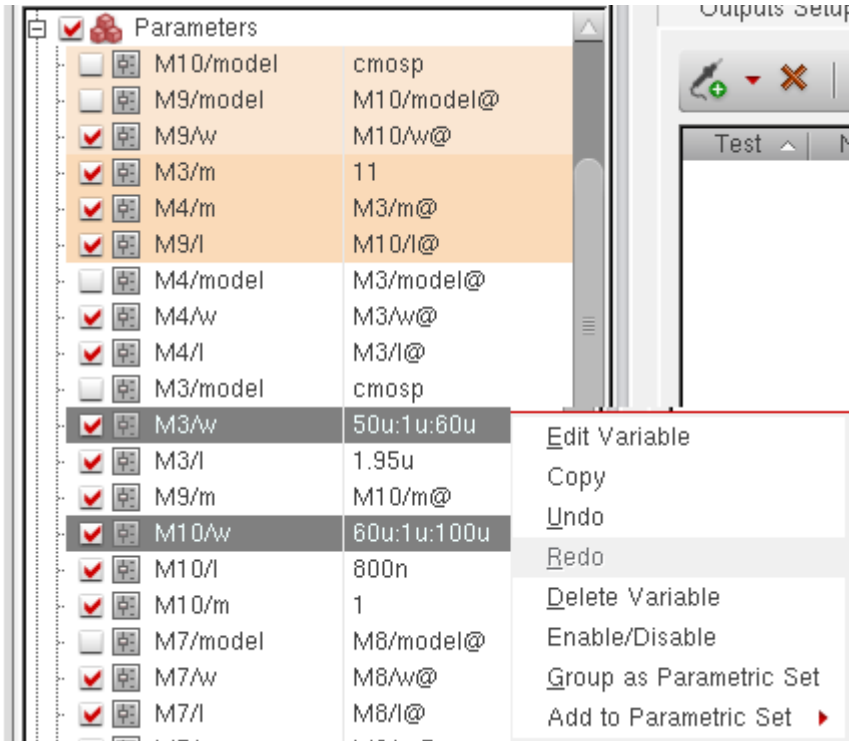
- The variables or parameters in a parametric set are bound as a group.
- When you create a parametric set, the list of variables is realigned to display all the variables or parameters of a group together.


© 2023 Cadence Design Systems, Inc.

- Each parametric set is highlighted with a different color, as shown below.



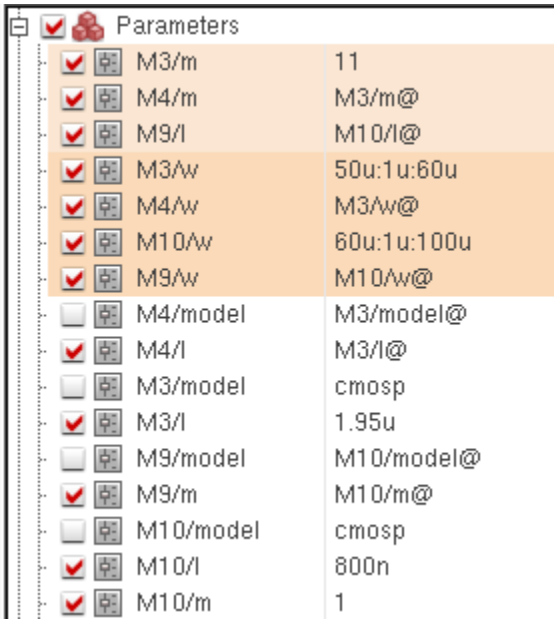

- When you group two or more device parameters together, the tool automatically identifies matched parameters and includes them in the parametric set. Consider an example. From the *Parameters* list, select two variables, `M3/w` and `M10/w` and add them to a parametric set, as shown below.





© 2023 Cadence Design Systems, Inc.

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 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 variables and parameters.
- If a parametric set is specified for global variables in ADE Assembler, then all the variables that are the part of that parametric set must have same number of variable values.

## Adding a Variable to a Parametric Set

To add a variable to an existing parametric set:

1. Right-click a variable and choose *Add to Parametric Set*.
2. From the list of existing parametric sets shown in the sub-menu, 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 and parameters are rearranged to display all variables or parameters of a group together.

**Note:** To add a parametric set for local variables in an existing maestro view, refer to the SKILL function, `ocnxlLocalParametricSet.`


© 2023 Cadence Design Systems, Inc.

## Removing a Variable from a Parametric Set

To remove a variable from a parametric set:

➡ Right-click 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 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 Variables or Parameters list.

## Enabling or Disabling Parametric Sets

You can disable or enable all parametric sets present for design variables in the Setup assistant.

To disable all parametric sets:

➡ Right-click *Design Variables* and choose *Disable All Parametric Sets*.

After the parametric sets are disabled, the variables are moved back to their original location in the list. If the parametric sets are disabled, you can enable all the sets together. To do this, right-click *Design Variables* and choose *Enable All Parametric Sets*. All the parametric sets become visible again.


© 2023 Cadence Design Systems, Inc.

# 6

# Working with 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 Explorer, 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 a test and measure how the test performs 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 Explorer.

See the following topics for more details:


- Opening the Corners Setup Form on page 542
- Adding Corners on page 546
- Modifying Corner Values on page 558
- Working with Corners on page 559
- Working with Corner Groups on page 582
- Filtering Corners on page 587
- Simulating Corner on page 595



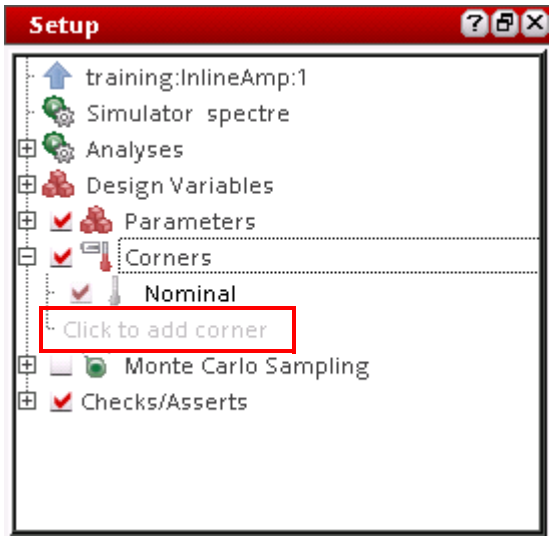
© 2023 Cadence Design Systems, Inc.

# Opening the Corners Setup Form

You can add or remove corners or modify corner settings by using the *Corners Setup* form.

Perform the following steps to open the Corners Setup form:

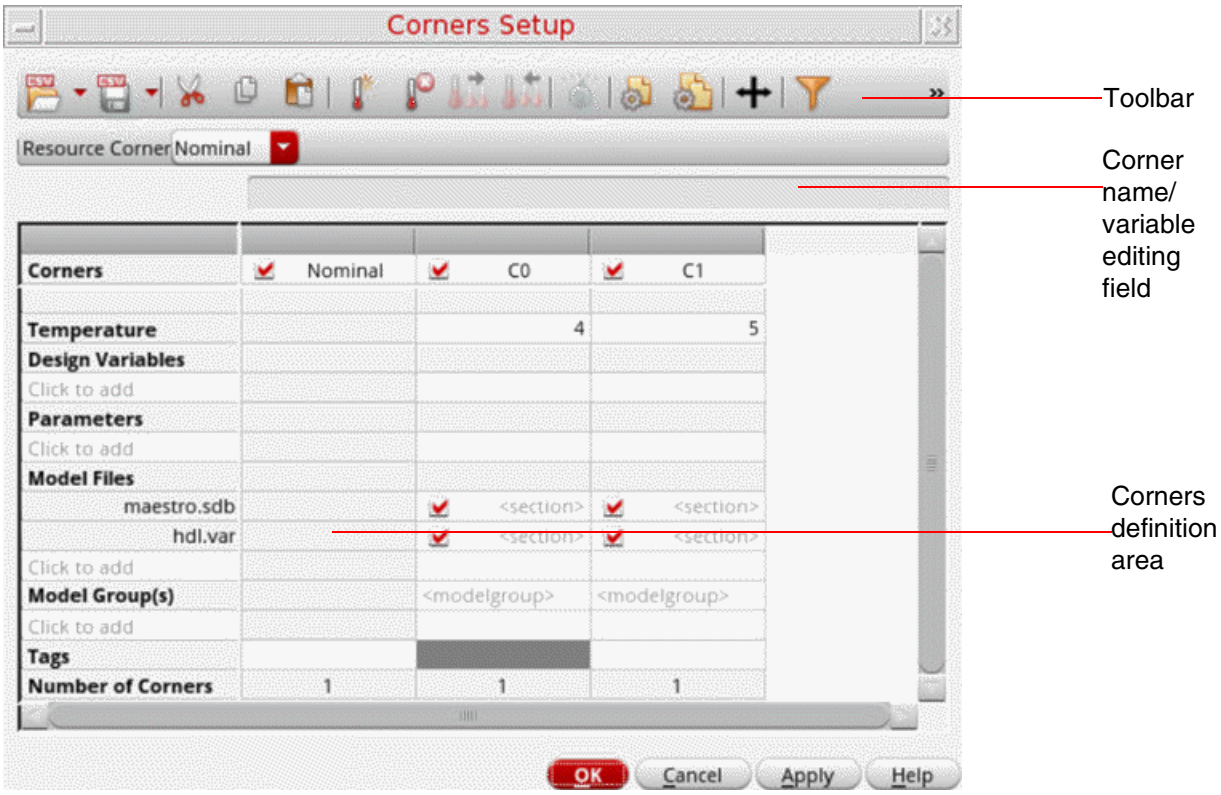
1. In the Setup assistant, expand *Corners* and click the text *Click to add corner*.





© 2023 Cadence Design Systems, Inc.

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 the following:

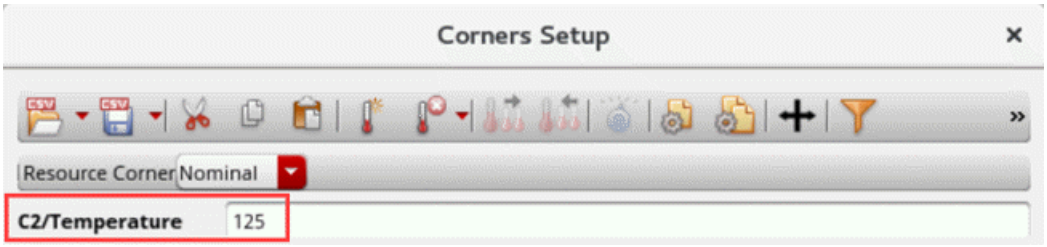
- A toolbar that provides the following commands:

| Toolbar button | Description |
|---|---|
| | Loads corners from a setup database file<br>Imports corners from a Process Customization Files (PCF) or Design Customization Files (DCF). |
| | Moves the selected text to the clipboard<br>Copies the selected text to the clipboard<br>Pastes the copied text from the clipboard<br>Creates a new corner |


© 2023 Cadence Design Systems, Inc.

| | |
|---|---|
| | Specifies which corners are to be removed. |
| *Remove Selected Corners* | Removes the highlighted corners. |
| *Remove Hidden Corners* | Removes the hidden corners. |
| *Remove Corners with Undefined Variables* | Removes the corners containing variables that have no value specified in a corner or the sections *Design Variables* and *Global Variables* of the Setup Assistant. |
| *Remove Duplicate Corners* | Removes any duplicate corners. These are corners that contain the exact same specified conditions. |
| | 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 |

- *Resource Corner* drop-down list — Lets you select the corner for which you want to assess the resource requirement. By default, the tool runs resource estimation on the nominal corner. If your setup contains corners that would require a higher level of resources than nominal, it is a better option to assess the resource requirement for a corner with the highest resource requirements. For details, refer to Running Resource Estimation for a Specific Corner.
- 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.



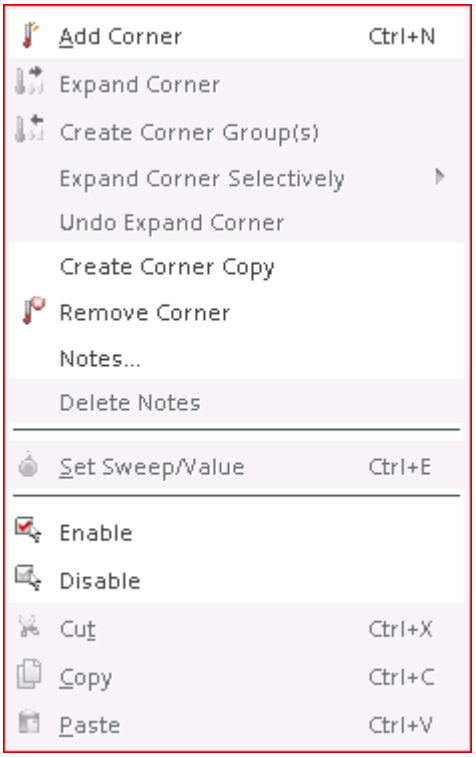

- Corner definition area — Displays the corner definitions in a tabular format. The first column in the table lists the variables organized into various groups. Each


© 2023 Cadence Design Systems, Inc.

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.

- Context-sensitive menu — The most frequently used commands for corners are available in the context-sensitive menu of the *Corners Setup* form. When you right-click in a corner column, the following context-sensitive menu is displayed:




For more details about how to add a corner or specify variables, refer to the following sections:

- Adding Corners on page 546
- Modifying Corner Values on page 558
- Working with Corners on page 559
- Working with Corner Groups on page 582
- Simulating Corner on page 595


© 2023 Cadence Design Systems, Inc.

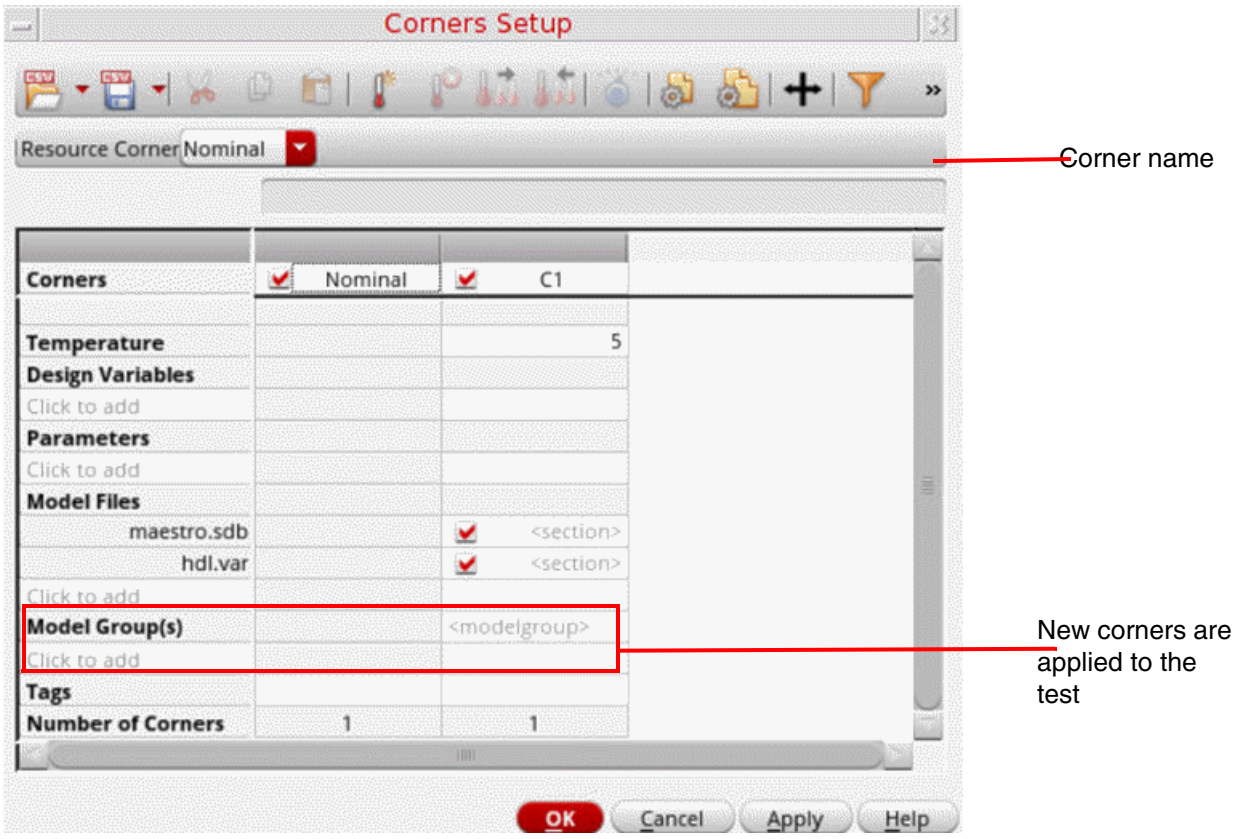
# Adding Corners

You can add a corner to define the conditions under which you want to check a design of your test.

Perform the following steps to add a new corner:

1. In the *Corners Setup* form, click *Add new corner* on the toolbar. Alternatively, right-click in an existing column and choose *Add Corner.*

   A new column is added in the *Corners Setup* form, as shown in the following figure.




   New corners are, by default, applied the test and are named sequentially, prefixed with `C`, such as `C1`, `C2`, and so on. You can change the names, if required.

2. 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*


© 2023 Cadence Design Systems, Inc.

*Variable*, are by default edited in the append mode. When you click 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.

**Note:** If you have saved a set of default corners in a `.sdb` file, to be used for an ADE Explorer view, set the `defaultCorners` environment variable to that file path. If no other corners are defined for the ADE Explorer view, corner definitions from the given file are loaded into the corners setup.

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 Setting up a Monte Carlo Simulation chapter.

***Related Topics***

Specifying Temperature

Specifying Values for Design Variables and Parameters

Specifying Values for Parameters

Adding Model Files to a Corner

Specifying Sections for Model Files

Creating a Model Group

Adding Corners Tags


© 2023 Cadence Design Systems, Inc.

## Specifying Temperature

To specify the temperature for a corner, in the column for that corner:

➡ Click the cell next to the *Temperature* label and type a value.

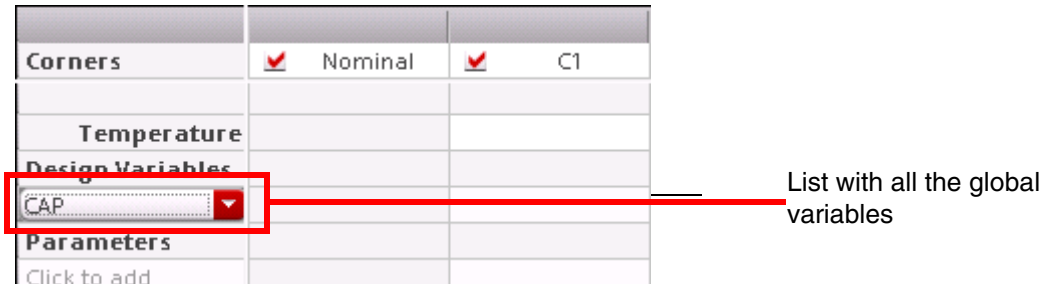
**Note:** You cannot specify a value if a parameter or variable named `temperature` is specified in your ADE Explorer test setup.

## Specifying Values for Design Variables and Parameters

To specify the value for a design variable for a corner, in the column for that corner:

1. Under *Design Variables*, click the text *Click to add*.

   A drop-down list containing all the global variables is displayed.



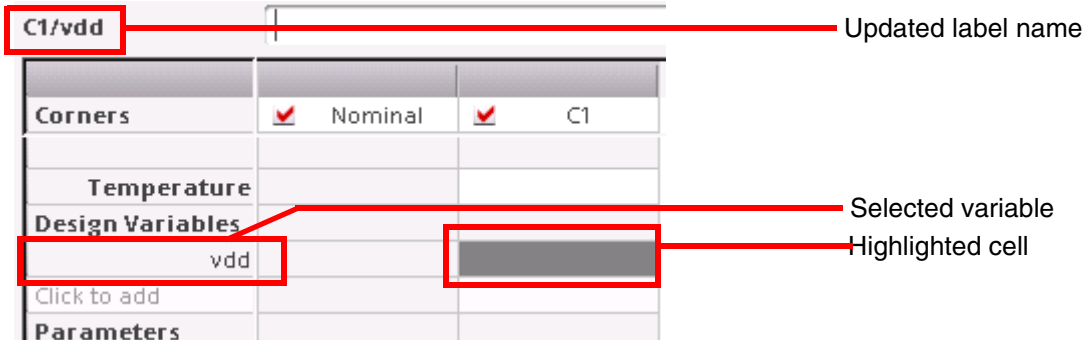

2. From this list, select the name of the variable which you want to add to the corner.
3. To specify the value for this variable, click in the cell corresponding to this variable in the column that contains the corner definition.

   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.


© 2023 Cadence Design Systems, Inc.

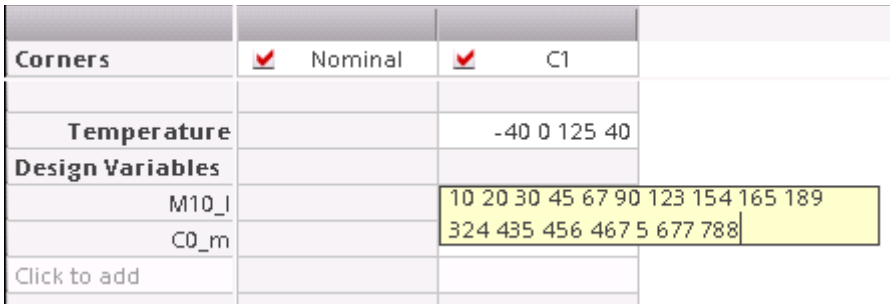
For example, when you select the `vdd` variable from the list, you will see the changes highlighted in the following figure.




4. Specify a 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.




   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 *Expand corner group* on the toolbar.

   **Note:** To remove the design variables for a corner, right-click the corresponding cell and select one of the following options:

   - *Remove*: Removes the selected design variables.
   - *Remove Unselected*: Removes the unselected variables.

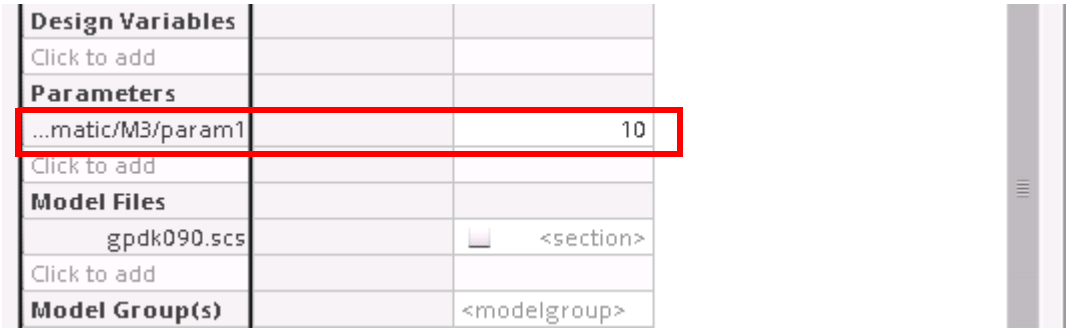

© 2023 Cadence Design Systems, Inc.

*Important*

By default, before running a simulation, ADE Explorer 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 Explorer setup, a redundant simulation is run for that variable. To ensure that any design variable that does not exist in the ADE Explorer setup is reported as an error before running a simulation, set the `showErrorForNonExistingVariables` environment variable to `t`.

### Specifying Values for Parameters

For a corner, you can specify parameters in the same way as you specify variables. When you click the text *Click to add* under *Parameters*, a drop-down list containing the names of all the CPF parameters, created for the cells or instances used in the test, is displayed. Select the required parameter from this list and specify a value for the parameter.



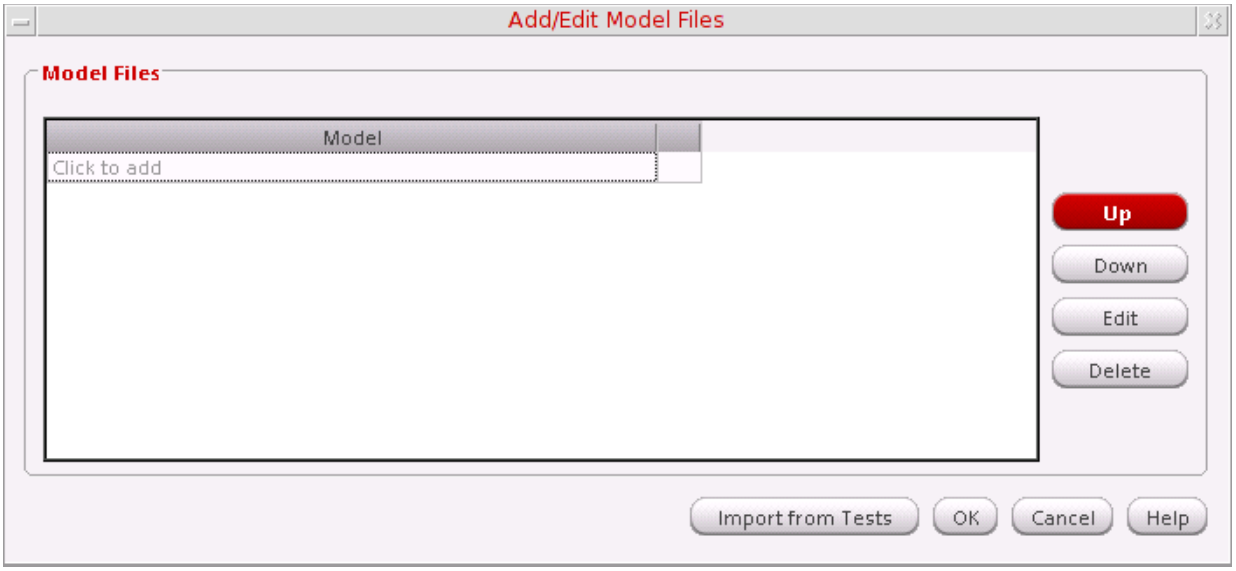

## Adding Model Files to a Corner

Perform the following steps in the *Corners Setup* form, for each model file you want to add as a corner:

1. In the *Corners Setup* form, perform one of the following steps:
   - On the toolbar, click *Add/Edit Model File(s)* .
   - Under *Model Files*, click the text *Click to add*.

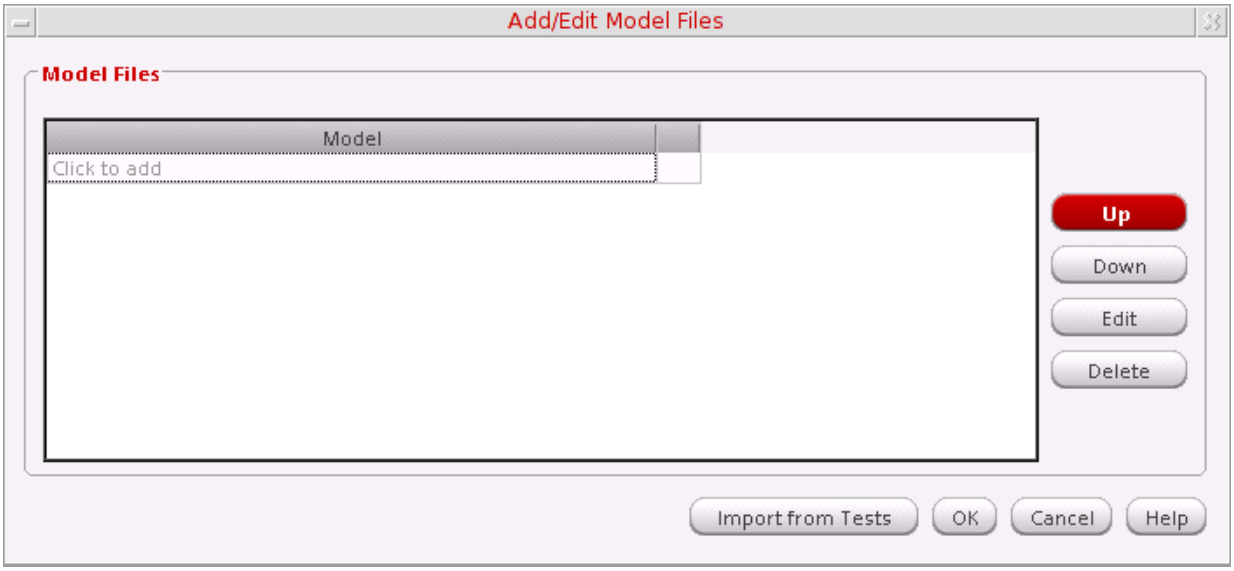

© 2023 Cadence Design Systems, Inc.

The *Add/Edit Model Files* form is displayed.




2. In the *Model Files* group box, do one of the following to add model files:
   - Click the text *Click to add* and type the path to and name of a model file.
   - Click the ellipses button to select a model file.
   - Click *Import from Tests* to import model files from the test in your setup.

     All the model files specified for test in your setup are displayed in the form. For more information about specifying model files for a test, see Specifying Model Libraries.





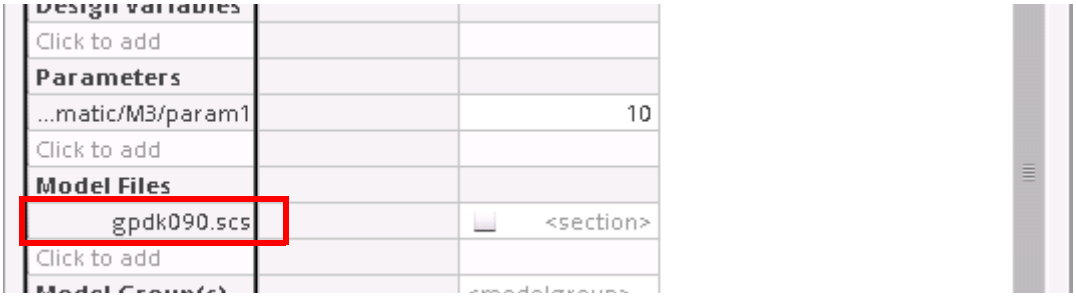
© 2023 Cadence Design Systems, Inc.

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:

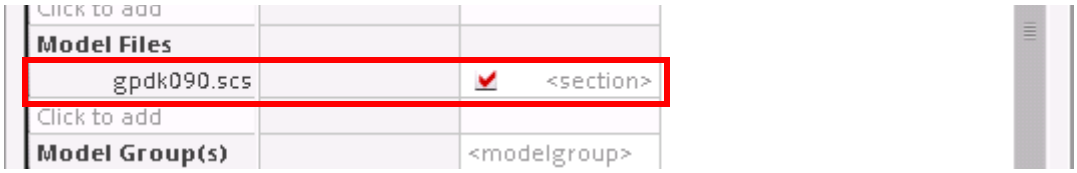
- Modify the search order which the simulator uses to find simulation models in the model files — By default, the simulator searches for models in the model files in the order as they are listed in the *Corners Setup* form. The simulator uses the first simulation model it finds with the matching name. 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.



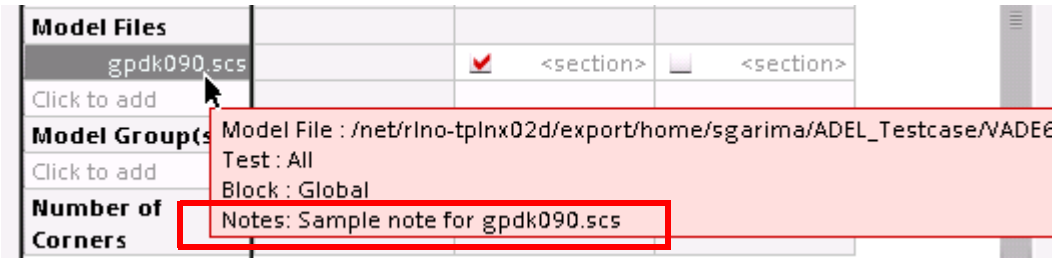

4. In the column for a corner, select the check box in the cell corresponding to the model file, to add it to the corner.





© 2023 Cadence Design Systems, Inc.

5. If required, add notes for the model file. To do this, right-click a model file and select *Notes*. In the *Add/Edit Notes* form, add notes for the model file and click *OK*. The notes added for model files are shown in their tooltips.




6. You can also view the model file in a text editor. To do this, right-click the model file and select *View Model File*.

7. 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, 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.

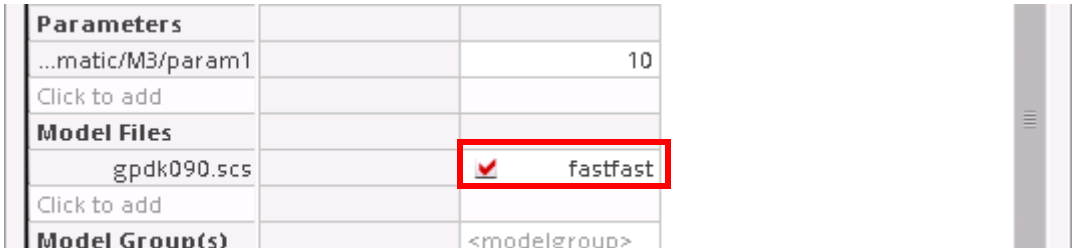
## Specifying Sections for Model Files

When you specify a model file for a corner in the column for that corner, the cell corresponding to the model file shows 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 containing the names of all the sections defined in the file is displayed. Select a section that


© 2023 Cadence Design Systems, Inc.

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 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.

## Adding a Model Group to a Corner

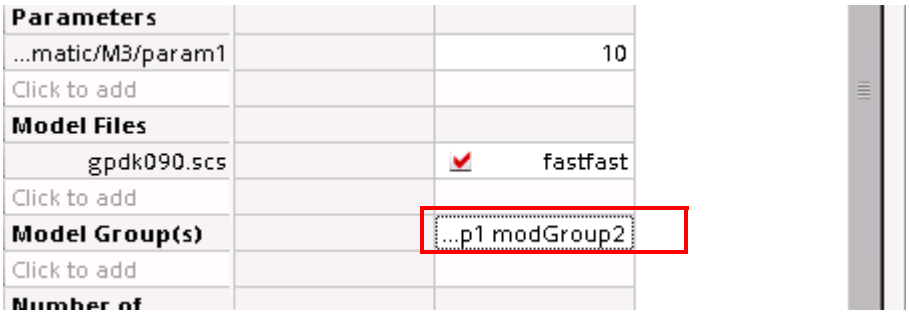
Perform the following steps in the *Corners Setup* form to add one or more model groups to a corner:

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 all the model groups which you want to add to the corner, one by one.

   Each model group that you select is displayed in a space-separated list in the column for the corner.





© 2023 Cadence Design Systems, Inc.

**Note:** For every model group that you add, a separate corner simulation will run.

See also:

- Creating a Model Group on page 555
- Saving and Loading Model Groups on page 557

## Creating a Model Group

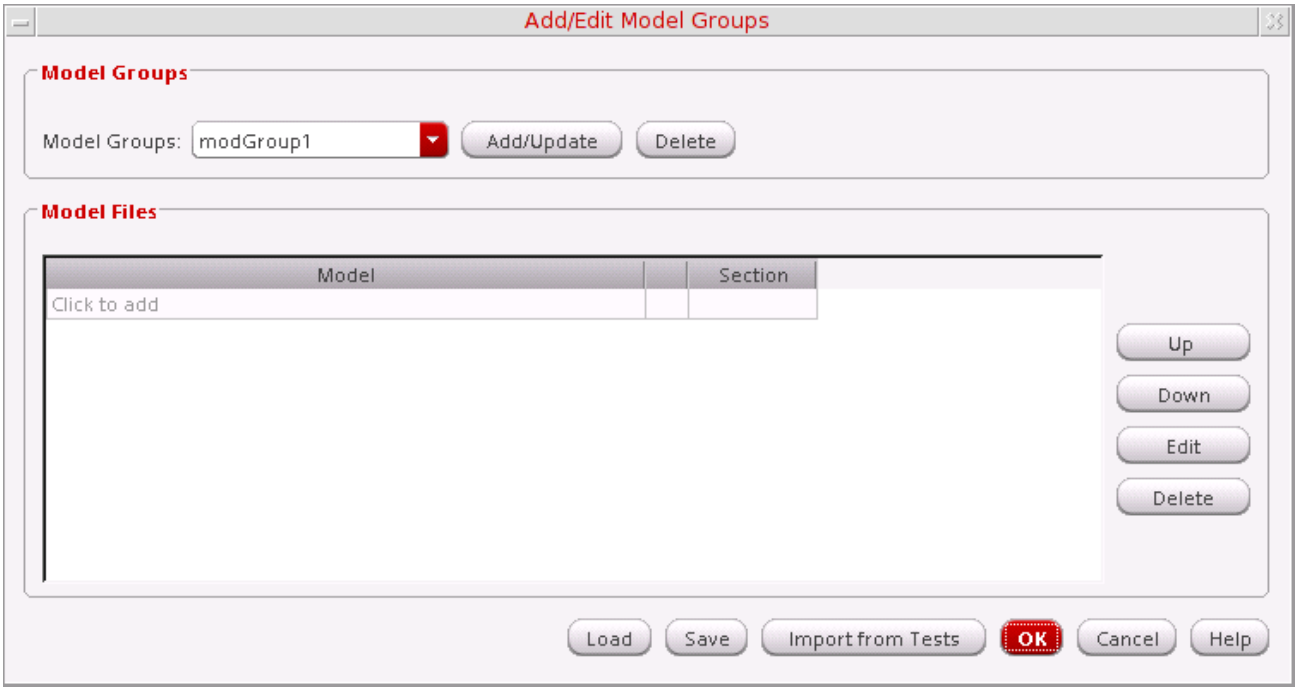
A model group contains model files that you can use when simulating a corner. For each model file in the group, you can also specify selected sections to be used.

Perform the following steps in the *Corners Setup* form to create a model group:

1. Click *Add/Edit Model Group(s)* on the toolbar.

   Alternatively, click the text *Click to add* under *Model Group(s)*.

   The *Add/Edit Model Groups* form is displayed.




   In the *Model Groups* field, type a name for the model group and click *Add/Update*.

2. Perform one of the following steps to add one or more model files to the model group:


© 2023 Cadence Design Systems, Inc.

- In the *Model* column of the *Model Files* table, click the text *Click to add* and type a valid path to a model file.
- In the *Model Files* table, click the ellipses 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 the test 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 clearing the check box to the left of the model file.

3. Optionally, 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.
4. 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 you specify and this name is also added to model groups drop-down list in the *Corners Setup* form.


© 2023 Cadence Design Systems, Inc.

## 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 Explorer view.

### Saving a Model Group

Perform the following steps to save model group information:

1. Click *Save*. The *Save Model Groups* form is displayed.
2. Locate the directory in which you want 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 XML format and the filename contains the `.sdb` extension.

### Loading a Model Group

Perform the following steps to load model group information from a previously saved file:

1. Click *Load*. The *Load Model Groups* form is displayed.
2. Locate the required model group information file.
3. Click *Open*.

The model group information is displayed in the *Add/Edit Model Groups* form.

## Adding Corners Tags

To filter or search through a long list of corners, you can assign specific tags to them.

To know more, see Assigning Tags to Corners.

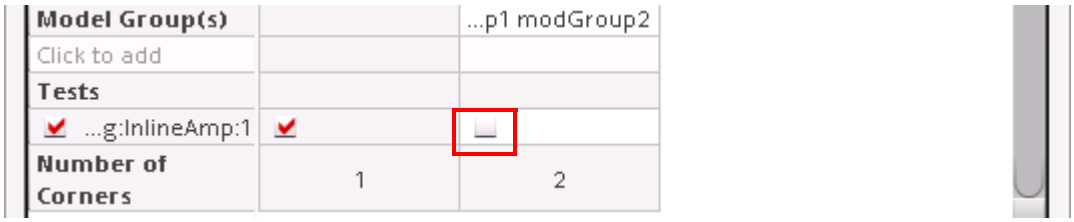

© 2023 Cadence Design Systems, Inc.

# Modifying Corner Values

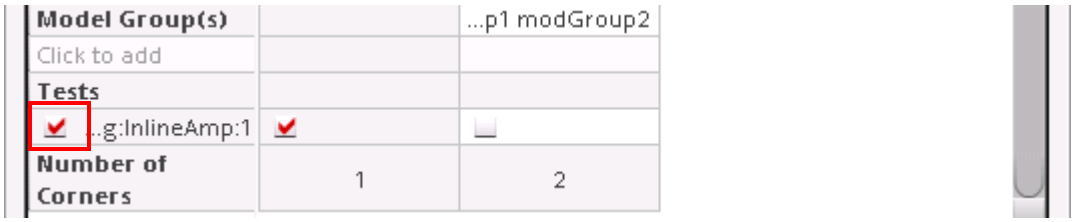
After you add corners, you can modify values for individual corners in the *Corners Setup* form. To edit the values of design variables or parameters, select the cell to display its value in the variable editing text field. You can edit the value in this field. Alternatively, double-click in the cell containing that value and edit the value there.

In addition, to editing values, you can also perform one of the following steps to modify a corner:

- Enable/disable a model for a corner— Select/deselect the check box next to the model name in the column for the corner.




- You can enable/disable all corners—Select/deselect the check box to the left of the test name.




- Remove a design variable, parameter, or model file from the *Corners Setup* form — 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`.

See Also:

- Disabling and Enabling One or More Corners on page 562
- Changing the Order of Corners on page 564


© 2023 Cadence Design Systems, Inc.

# Working with Corners

This section describes the various tasks that you can perform while working with the corners.

- Renaming a Corner on page 559
- Adding Notes to a Corner on page 560
- Disabling and Enabling Corners on page 560
- Changing the Order of Corners on page 564
- Removing Corners Using the Setup Assistant on page 565
- Removing Corners Using the Corners Setup Form on page 565
- Viewing Corner Settings on page 569
- Copying Corners on page 569
- Exporting Corners on page 570
- Importing Corners on page 573
- Importing Corners from Customization Files on page 578
- Setting Up a Default Set of Corners on page 580
- Running Resource Estimation for a Specific Corner on page 581

## Renaming a Corner

By default, the program names corners as `C1`, `C2`, and so on.

Perform the following steps to change the default name of a corner:

1. In the Corners Setup form, select the name of the corner.

   The name of the corner is displayed in *Corner Name* text field.
2. Change the name of the corner and press *Enter*.
3. Click *Apply.*


© 2023 Cadence Design Systems, Inc.

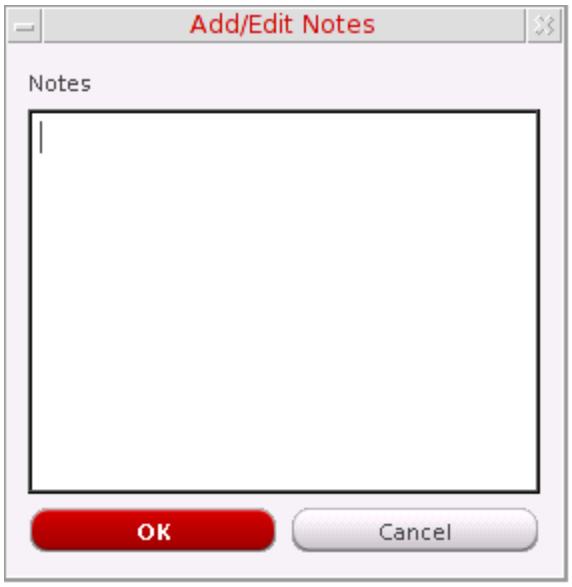
## Adding Notes to a Corner

If you have some additional information to add to the corners, you can add notes for each individual corner, which you can refer to later.

Perform the following steps in the Setup assistant or in the Corners Setup form to add notes for a corner:

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.

## Disabling and Enabling Corners

By default, a corner is enabled to run for the test in the current ADE Explorer setup. You can selectively disable and enable the nominal corner, one or more corners for the test, or all corners.

- Disabling and Enabling the Nominal Corner on page 561
- Disabling All Corners on page 561


© 2023 Cadence Design Systems, Inc.

- Disabling and Enabling One or More Corners on page 562
- Changing the Order of Corners on page 564

### Disabling and Enabling the Nominal Corner

Perform the following step to disable or enable the nominal corner simulation:

➡ Deselect/select the check box next to the test name in the column for the nominal corner.

Perform one of the following steps to disable or enable the nominal corner simulation:

➡ In the *Corners* section on the Setup assistant, deselect/select the *Nominal* check box.

➡ In the *Corners Setup* form, deselect/select the *Nominal* check box.

**Note:** After running the simulation, only the columns for enabled corners are created in the *Results* tab.

### Disabling All Corners

Perform one of the following step to disable all corners:

➡ In the *Corners Setup* form, right-click a test name and choose *Disable Corners on Test*.

The check boxes for all the corners, except the nominal corner, are cleared. The simulator does not run the corner simulations for the test.

➡ In the Setup assistant, deselect the check box next to the *Corners* section.

All the corners, including the nominal corner, are disabled. The simulator does not run the corner simulations for the test.

### Enabling All Corners

Perform one of the following step in the *Corners Setup* form to enable all corners:

➡ In the *Corners Setup* form, right-click the test name and choose *Enable Corners on Test*.

The check boxes for all the corners are selected. The simulator runs the corner simulations for the test.

➡ In the Setup assistant, select the check box next to the *Corners* section.


© 2023 Cadence Design Systems, Inc.

All the corners are enabled. The simulator runs the corner simulations for the test.

### Disabling and Enabling One or More Corners

Perform the following steps in the *Corners Setup* form to disable one or more corners:

1. Select a corner.
2. Optionally, 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 an enabled corner and choose *Disable*.

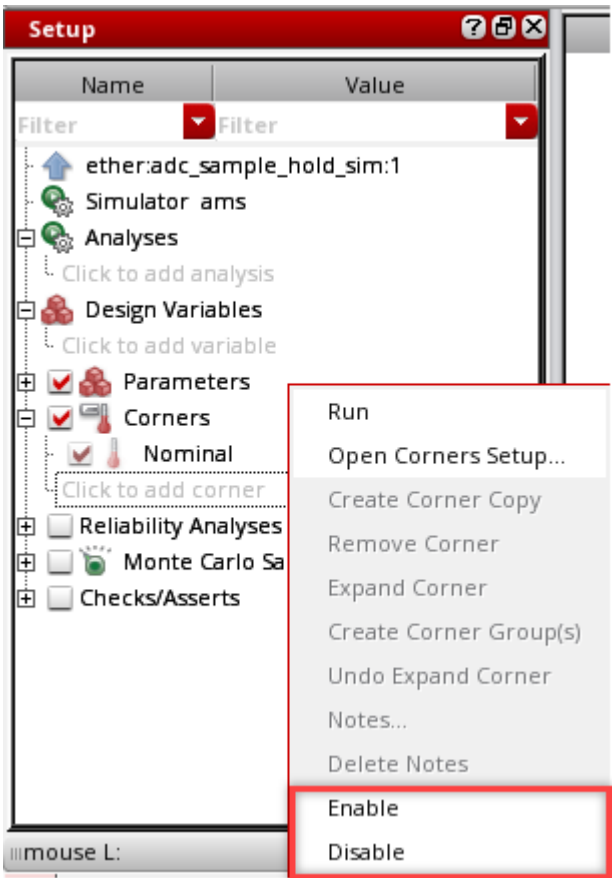
   The check boxes next to the names of all the selected corners are cleared. The simulator does not run those corner simulations.

To enable one or more corners, select the name of the corners as described above, and then right-click and choose *Enable*.

Alternatively, to enable or disable one or more corners in the *Setup* Assistant, select the


© 2023 Cadence Design Systems, Inc.

name of the corners, right-click and choose *Enable* or *Disable*.




**Note:** After running the simulation, only the columns for enabled corners are created in the *Results* tab.


© 2023 Cadence Design Systems, Inc.

## Changing the Order of Corners

You can change the order in which the corners are listed in the Setup assistant or the *Corners Setup* form. Any change done in either of these is reflected in the other as well. This also defines the order in which the corners are simulated.

You can change the order of corners either in the *Corners Setup* form or in the Setup assistant.

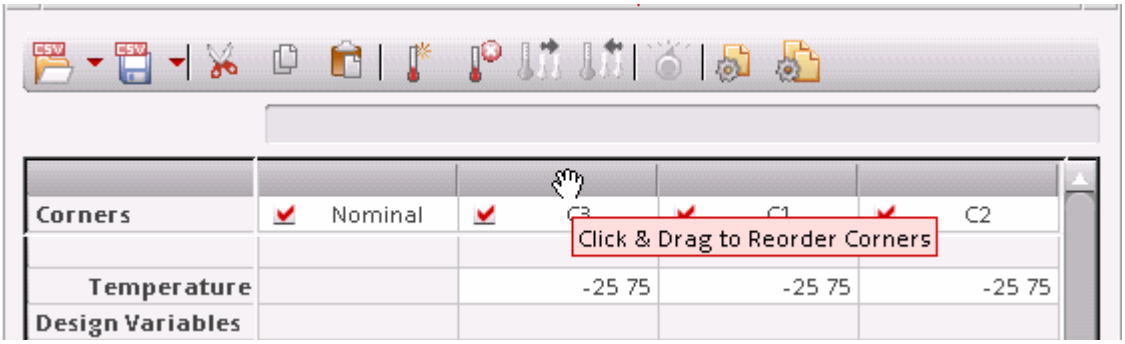
Perform the following steps to change the order of a corner in the Setup assistant:

1. Expand *Corners* to view all the corners.
2. Click a corner name and drag it to the desired location.

   A blue indicator line appears at the location where you have moved it. The tooltip also shows the name of the corner being moved.
3. Drop the corner.

Perform the following steps to change the order of a corner in the *Corners Setup* form:

➡ Click the gray column header for a corner, drag and drop the column to a desired location.




### *Important Points to Note*

- To move multiple corners at a time, press the *Control* key and select all the corners, and then drag them to the new location.
- You cannot change the order for the `Nominal` corner.
- In the *Corners Setup* form, it is not possible to drag and drop a corner to the right-most location.


© 2023 Cadence Design Systems, Inc.

## Removing Corners Using the Setup Assistant

Perform the following steps in the Setup assistant to remove one or more corners:

1. Expand *Corners* and select the name of the corner you want to remove.

   Optionally, Use the `Shift` key (for contiguous corners), or the `Control` key (for noncontiguous corners) and click the next corner to add more corners to the selection set.

2. Right click the corner name and choose *Remove Corner*.

   The selected corners are removed from the *Corners* section.

## Removing Corners Using the Corners Setup Form

Perform the following steps in the *Corners Setup* form to remove one or more corners:

1. Select the name of a corner.

   Optionally, use the `Shift` key (for contiguous corners), or the `Control` key (for noncontiguous corners) and click the next corner to add more corners to the selection set.

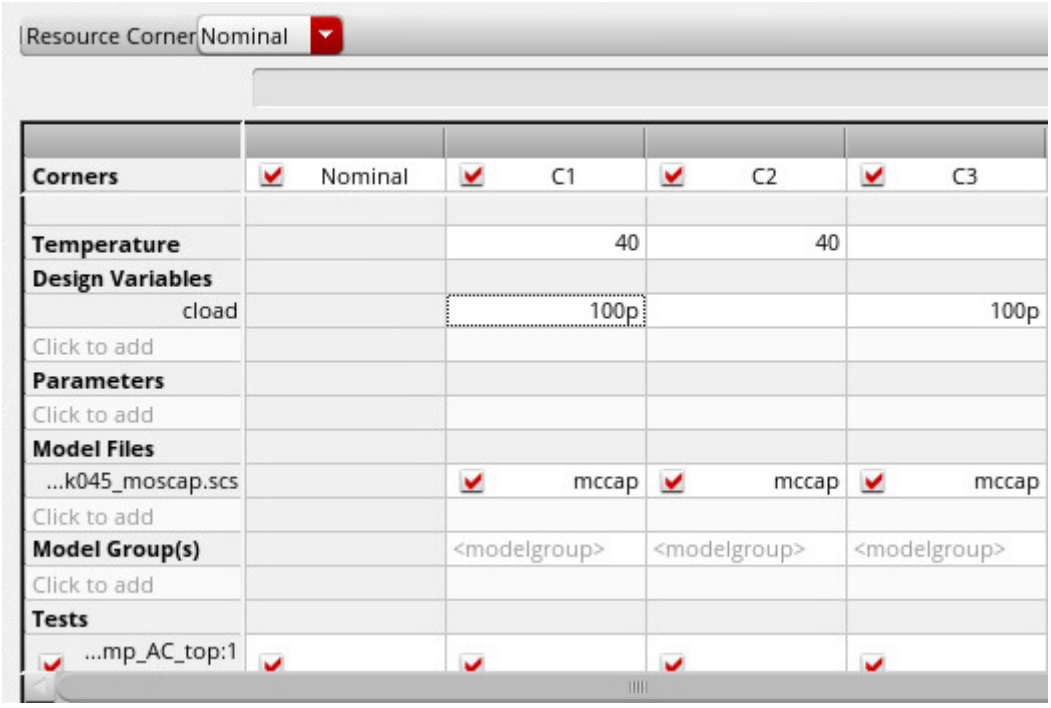
2. Select one of the following options from the *Remove Corner* command on the toolbar.

   - *Remove Selected Corners:* Removes the selected corners.
   - *Remove Hidden Corners*: Removes the hidden corners
   - *Remove Corners with Undefined Variables:* Removes the corners containing variables that have no value specified in a corner or the sections *Design Variables* and *Global Variables* of the Setup assistant.

     For example, on selecting this option for the following setup, the corner `C2` gets removed since the variable `cload` does not have a value specified for it and is not


© 2023 Cadence Design Systems, Inc.

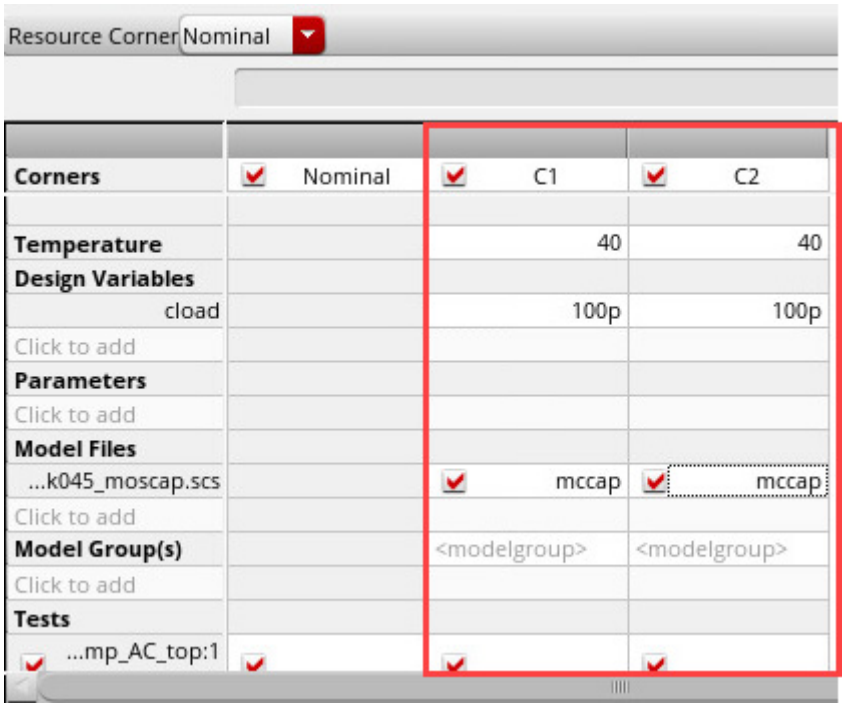
defined in the *Design Variables* or *Global Variables* section of the Setup Assistant.




- *Remove Duplicate Corners:* Removes any duplicate corners. These are corners that contain the exact same specified conditions.


© 2023 Cadence Design Systems, Inc.

For example, on selecting this option for the following setup, the corner `C2` gets removed since it contains the same specified conditions as `C1`.




The relevant corner columns are removed from the form. The number of corners reflected on the Setup assistant also decreases by the number of corners you remove.

## Managing Corners Using Editing Options

In case you have a standard set of instructions to review and consolidate a set of corners in the setup, you can specify these instructions as editing options.

To do this:

1. Set the function `maeSetCornerEditingOptions`.
2. Whenever you want to review the desired list of corners, set the function `maeApplyCornerEditingOptions`. This filters the list of corners according to the predefined settings.

ADE Explorer and ADE Assembler also provide the following functions:

- `maeGetCornerEditingOptions`: Lets you review the editing options specified through `maeSerCornerEditingOptions`.


© 2023 Cadence Design Systems, Inc.

- `maeResetCornerEditingOptions`: Resets the corners setup to its state before `maeSetCornerEditingOptions` was set.

### *Related Topics*

maeApplyCornerEditingOptions

maeGetCornerEditingOptions

maeResetCornerEditingOptions

maeSetCornerEditingOptions

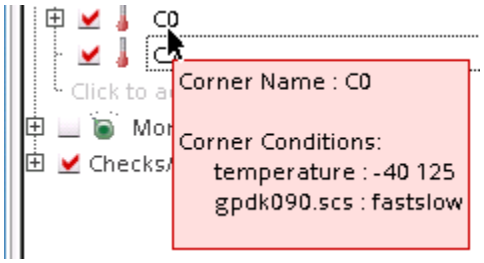

© 2023 Cadence Design Systems, Inc.

## Viewing Corner Settings

Perform the following step to view the settings for a corner:

➤ In the *Corners* section on the Setup assistant, hover the mouse pointer over the name of a corner.

The tooltip displays the corner settings, as shown in the following figure:




## Copying Corners

You can reuse the information in a corner by creating a copy of the corner.

Perform one of the following steps to create a copy of a corner:

➥ In the Setup assistant, expand *Corners*, right-click the corner name and choose *Create Corner Copy*.

A new corner containing the same corner information is added to the *Corners* section.

➥ Alternatively, in the *Corners Setup* form, 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 595.


© 2023 Cadence Design Systems, Inc.

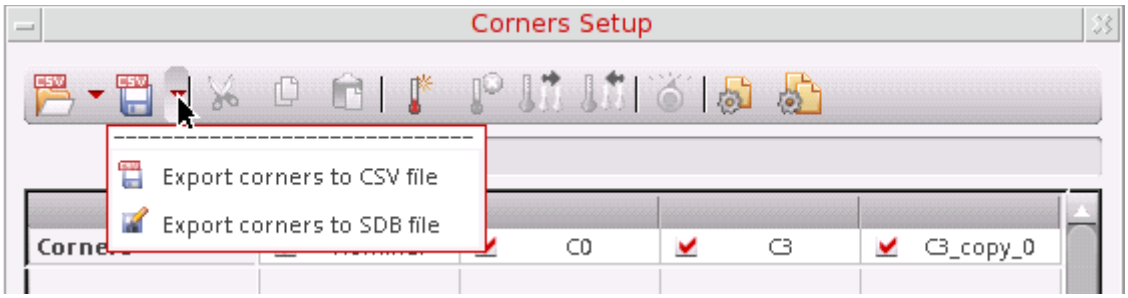
## Exporting Corners

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

Perform the following steps to export corners setup information in any of these formats:

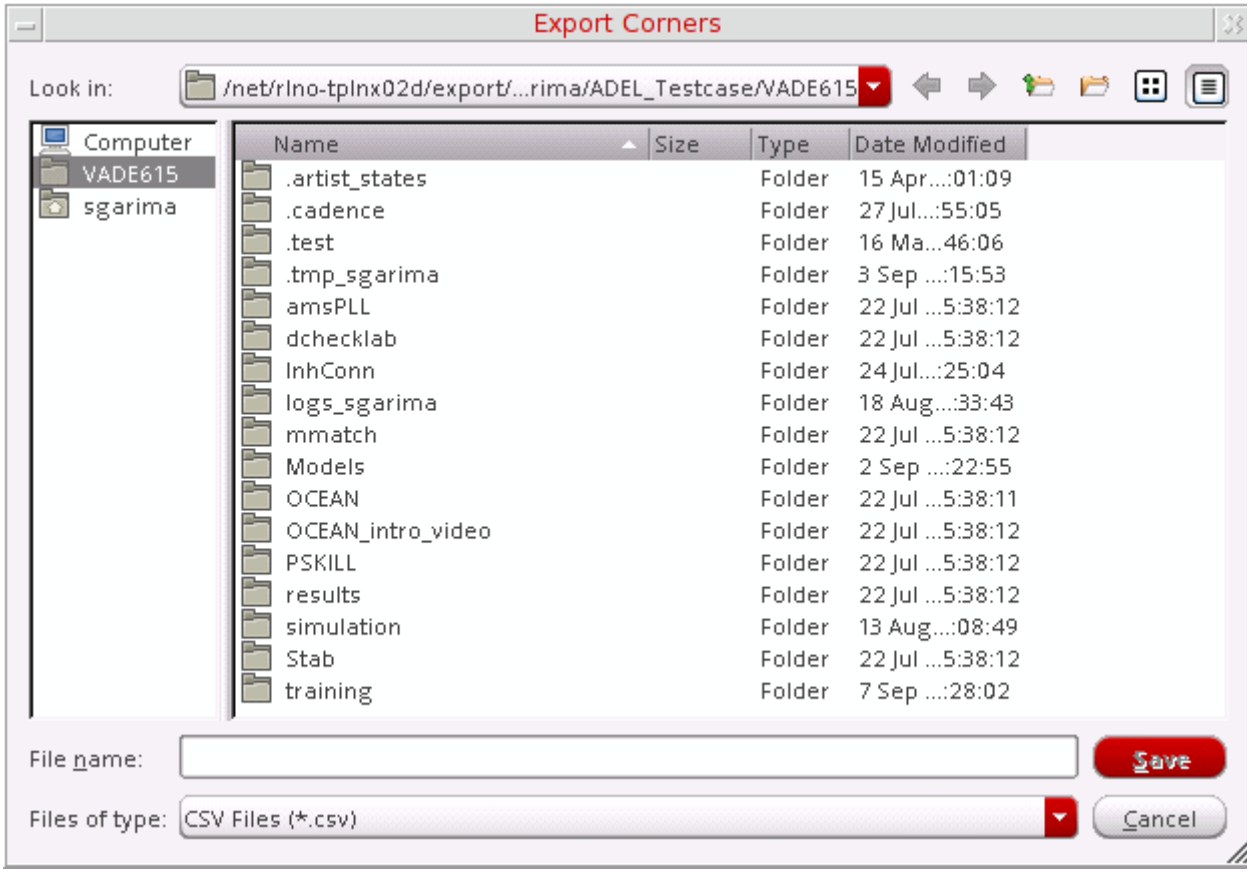
1. In the Corners Setup form, choose an appropriate command to export the corners, as shown in the figure below.




**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.


© 2023 Cadence Design Systems, Inc.

The *Export Corners* form is displayed, as shown below.



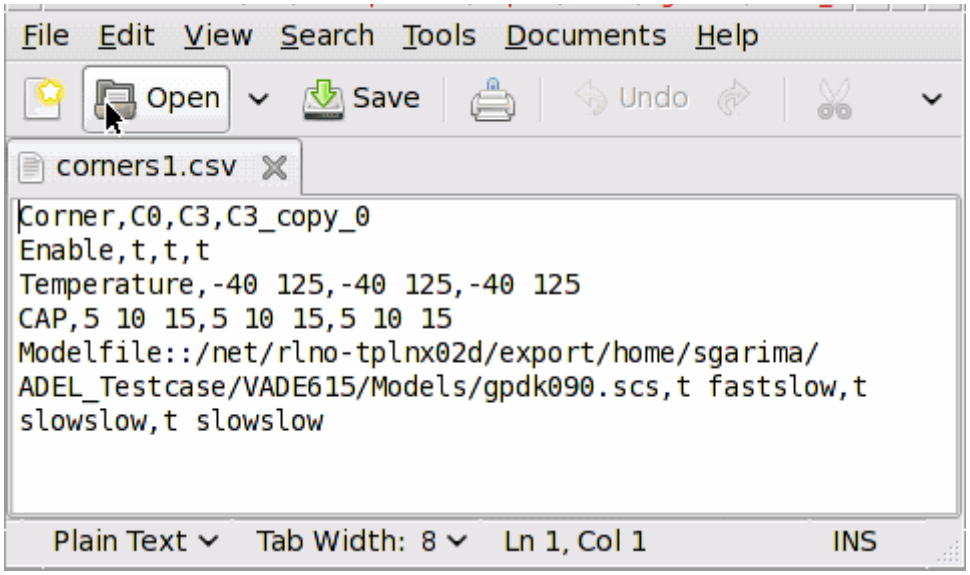

**Note:** The file extension displayed in the *Files of type* drop-down list box is determined by the export command that you select. If required, you can change the file type.

2. Optionally, browse and select a directory 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.


© 2023 Cadence Design Systems, Inc.

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 Explorer 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 Explorer does not save these in the `.csv` file and displays an appropriate message.


© 2023 Cadence Design Systems, Inc.

## Importing Corners

You can import the corner details from an existing `.csv` or `.sdb` or a customization file.

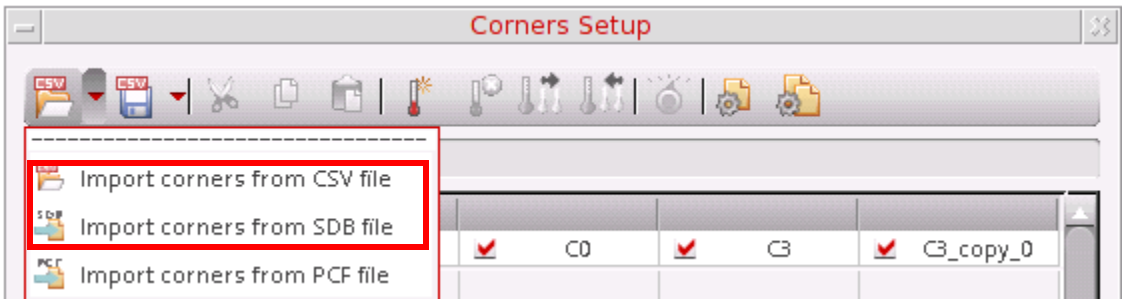
The following topics explain how to import corners from these files:

- Importing Corners from .csv or .sdb Files
- Importing Corners from Customization Files

### Importing Corners from .csv or .sdb Files

Perform the following steps to import the corners setup from a `.csv` or `.sdb` file:

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*.


© 2023 Cadence Design Systems, Inc.

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.

While importing the corners setup, if you have a predefined set of instructions specified through the function `maeSetCornerEditingOptions`, the tool reviews the corners according to these settings and filters the desired list of corners.

*Important*

When you import the corner details from a `.csv` file, ADE Explorer ignores the global variables and parameters that are not found in the list of global variables and parameters in the Setup assistant. A corresponding warning message is displayed in the CIW.

### *Related Topics*

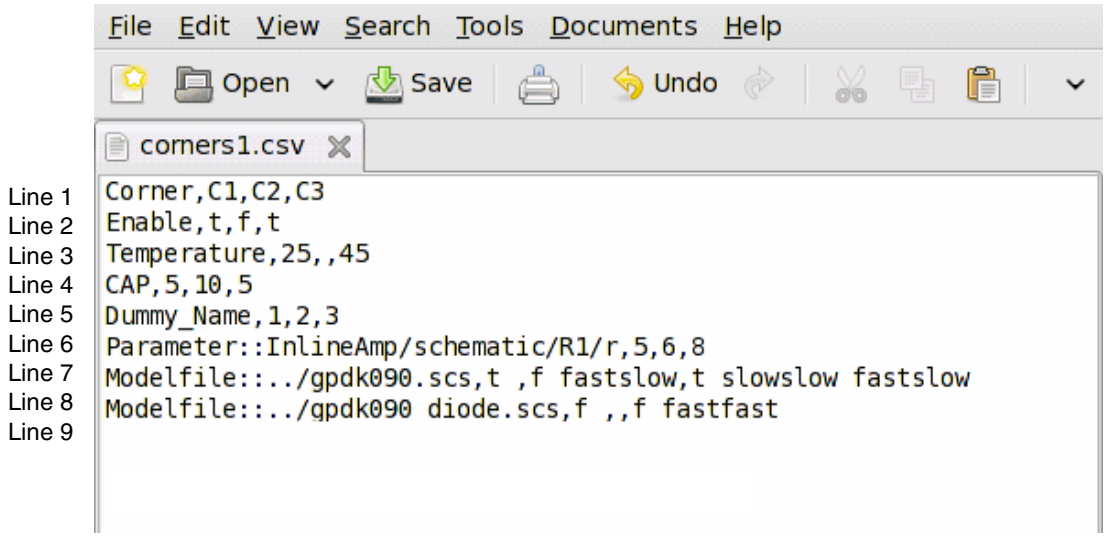
maeSetCornerEditingOptions

### *Format Details of a Corners CSV File*

While importing corner details from a CSV file, ADE Explorer 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 Explorer into the *Corners Setup* form.


© 2023 Cadence Design Systems, Inc.

The figure given below shows a sample CSV file.




In the figure above, 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 Explorer 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 test 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*


© 2023 Cadence Design Systems, Inc.

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 slowslow fastslow`, mentions that the model file has been enabled for corner `C3` and sections `slowslow` and `fastslow` 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** — The usage of corners for the test 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:InlineAmp:1,t,f,t
  ```

  `t` in the beginning indicates that corners will be run for the `Test::training_new_4:InlineAmp:1` test. The `t,f,t` list indicates 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.


© 2023 Cadence Design Systems, Inc.

### *Format of a .SDB File*

While importing corner details from a SDB file, the maestro cellviews require the corner details to be given in the required set of XML tags that it uses to identify different elements of a corner. This section explains the XML tags used to read corners:

- `<setupdb>`: **(Mandatory tag)** Specifies the name of the source setup database. The version property of this tag specifies the version of XML tags used in the .sdb file. The latest version is 6.
- `<active>`: **(Mandatory tag)** Specifies that the corners are to be imported for the currently active setup in the maestro view.
- `<corners>`: **(Mandatory tag)** Contains details for one or more corners. Each corner is specified by the `<corner>` tag.
- `<corner>`: Specifies the details for one corner. This tag can further contain `<vars>`, `<parameters>` or `<models>` tags to specify various corner settings.

### Example 1:

The following code shows an example input file that contains only one corner with minimum details to be imported.

```
<?xml version="1.0"?>
     <setupdb version="6">SDB_NAME
          <active>Active Setup
          <corners>
               <corner enabled="1">CORNER_NAME</corner>
          </corners>
          </active>
</setupdb>
```

### Example 2:

The following example input file contains multiple settings for the corners to be imported:

```
<?xml version="1.0"?>
     <setupdb version="6">b
     <active>Active Setup
     <corners>
          <!-- Can contain multiple corner tags-->
          <corner enabled="1">_default</corner>
          <corner enabled="1">C0
               <vars>
               <!-- Can contain multiple var tags to define corner variables -->
                    <var>temperature
                         <value>27</value>
                    </var>
                    <var>v1
                         <value>1</value>
                    </var>
               </vars>
               <parameters>
               <!-- Can contain multiple location tags to define different device
parameters in corners. Specify the location in the
```


© 2023 Cadence Design Systems, Inc.

```
libraryName/cellname/viewName/instName format-->
                    <location>design_180/inv/schematic/PM0
                        <parameter>fingers
                            <value>1u</value>
                        </parameter>
                    </location>
                </parameters>
                <models>
                    <!-- Can contains multiple model tags to define model files.
modelblock is used for MTS model setting. For non-MTS blocks, set it to Global.
modeltest is used to specify the test name for a model -->
                    <model enabled="1">gpdk.scs
                        <modeltest>All</modeltest>
                        <modelblock>Global</modelblock>
                        <modelfile> ./gpdk.scs</modelfile>
                        <modelsection>"FS"</modelsection>
                    </model>
                </models>
                <disabledtests>
                <!-- Defines the tests disabled for this corner. -->
                    <test>MTS_test2:testbench:1</test>
                </disabledtests>
            </corner>
    </corners>
    </active>
    </setupdb>
```

### Importing Corners from Customization Files

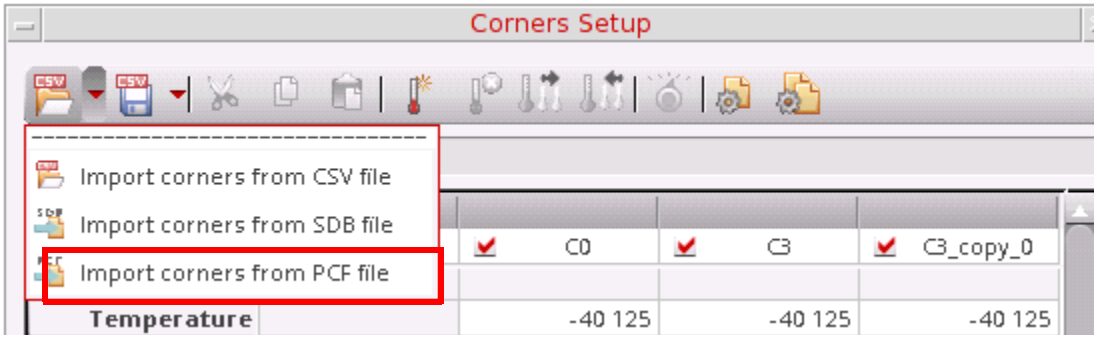
You can import a set of predefined corners and their measurements from the following types of customization files:

- 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.

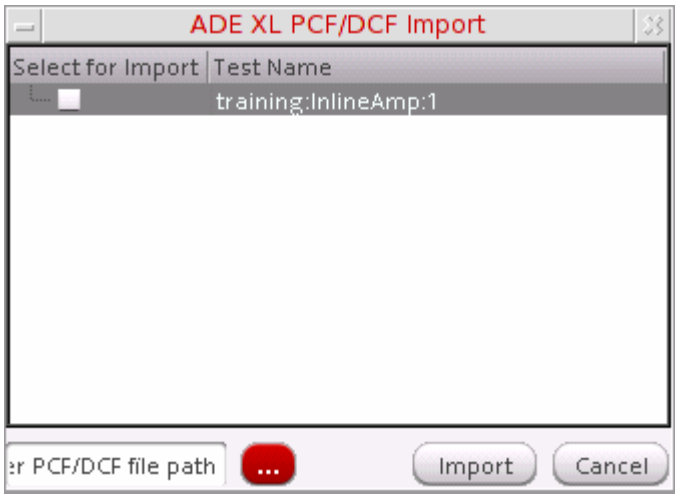

© 2023 Cadence Design Systems, Inc.

Perform the following steps for each customization file you want to import:




1. In the *Corners Setup* form, choose the *Import Corners from PCF File* command on the toolbar.

   The *ADE XL PCF/DCF Import* form is displayed.




2. Click the ellipses button.

   The *Choose PCF/DCF File for Import* form is displayed.

3. Browse and select the PCF/DCF file from which 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 the test name for which you want to use the imported corners.


© 2023 Cadence Design Systems, Inc.

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

Perform the following steps to automatically setup a default set of corners when you open an ADE Explorer view:

1. Use the procedure described in Exporting Corners on page 570 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 Explorer 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 Explorer view.

   You can use the default set of corners as is, or use them to quickly setup up corners for your test.

*Important*

The default corners will be loaded only if no other corner is defined in the *Corners Setup* form in the ADE Explorer view.

The following examples describe the use of the `adexl.gui defaultCorners` environment variable.


© 2023 Cadence Design Systems, Inc.

### 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)
```

## Running Resource Estimation for a Specific Corner

The *Resource Corner* drop-down list in the Corners Setup form lets you select the corner for which you want to assess the resource requirement. By default, the tool runs resource estimation on the nominal corner. If your setup contains corners that would require a higher level of resources than nominal, it is a better option to assess the resource requirement for a corner with the highest resource requirements. For details, refer to Running Resource Estimation for a Specific Corner.


© 2023 Cadence Design Systems, Inc.

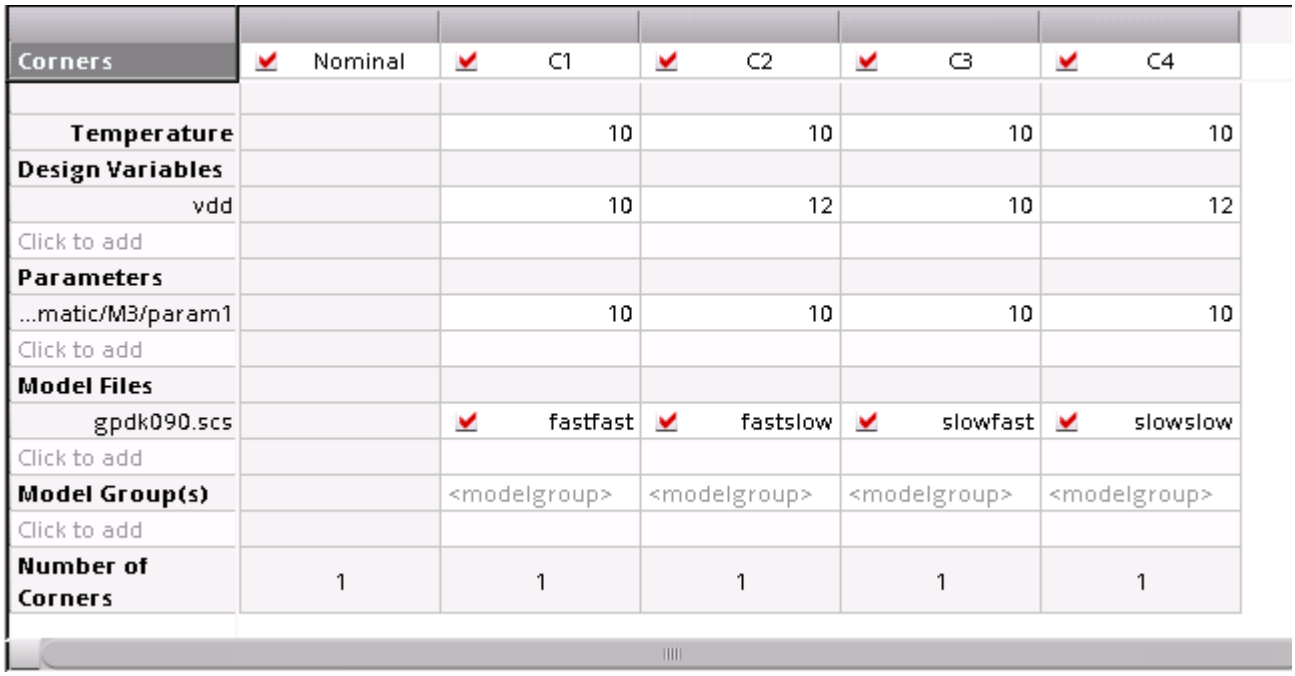
# Working with Corner Groups

A corner group is a set of corners that ADE Explorer displays as a single corner. For more information about working with corner groups, see the following topics:

- Creating a Corner Group on page 582
- Expanding a Corner Group on page 585

## 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 | | 10 | 12 | 10 | 12 |
| Click to add | | | | | |
| **Parameters** | | | | | |
| ...matic/M3/param1 | | 10 | 10 | 10 | 10 |
| Click to add | | | | | |
| **Model Files** | | | | | |
| gpdk090.scs | | fastfast | fastslow | slowfast | slowslow |
| Click to add | | | | | |
| **Model Group(s)** | | <modelgroup> | <modelgroup> | <modelgroup> | <modelgroup> |
| Click to add | | | | | |
| **Number of Corners** | 1 | 1 | 1 | 1 | 1 |

You can create groups for these corners, using the *Corners Setup* form or the Setup assistant, so that the *Corners Setup* form displays these corners in a single column. This helps in a better management of these corners.


© 2023 Cadence Design Systems, Inc.

### Creating Corner Groups Using the Corners Setup Form

Perform the following steps to create a corner group using the *Corners Setup* form:

1. 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 *Control* key (for noncontiguous corners) and click the next corner name to add more corners to the selection set.
2. Click *Create corner group* on the toolbar.

   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.

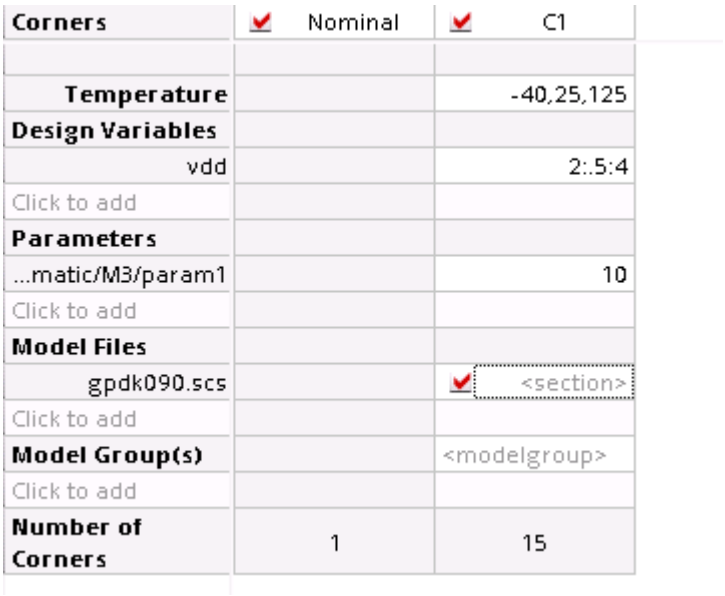
| Corners | Nominal | C1_0 | C1_1 |
|---|---|---|---|
| | | | |
| Temperature | | 10 | 10 |
| Design Variables | | | |
| vdd | | 10 | 12 |
| Click to add | | | |
| Parameters | | | |
| ...matic/M3/param1 | | 10 | 10 |
| Click to add | | | |
| Model Files | | | |
| gpdk090.scs | | ...st slowfast | ...w slowslow |
| Click to add | | | |
| Model Group(s) | | <modelgroup> | <modelgroup> |
| Click to add | | | |
| Number of Corners | 1 | 2 | 2 |

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.


© 2023 Cadence Design Systems, Inc.

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 corner `C1` 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.



| Corners | Nominal | C1 |
|---|---|---|
| | | |
| Temperature | | -40,25,125 |
| Design Variables | | |
| vdd | | 2:.5:4 |
| Click to add | | |
| Parameters | | |
| ...matic/M3/param1 | | 10 |
| Click to add | | |
| Model Files | | |
| gpdk090.scs | | <section> |
| Click to add | | |
| Model Group(s) | | <modelgroup> |
| Click to add | | |
| Number of Corners | 1 | 15 |

The program 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 Setup Assistant

Perform the following steps to create a corner group using the Setup assistant:

1. In the *Corners* section on the Setup assistant, select the names of the corners you want to include in the corner group.

   Hold down the *Shift* key (for contiguous corners) or the *Control* 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).


© 2023 Cadence Design Systems, Inc.

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

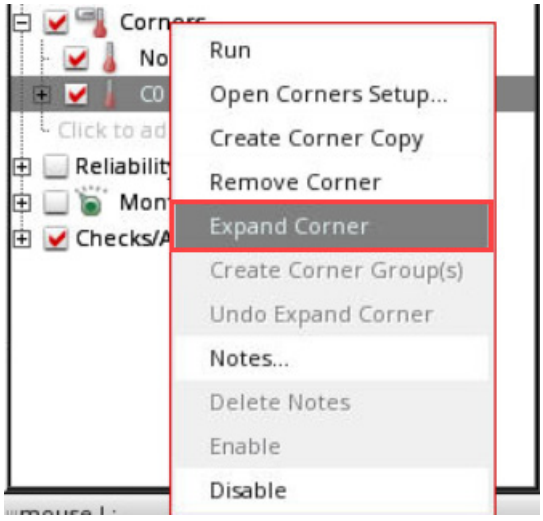
This section describes, how to:

- 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

Perform one of the following steps to expand a corner group, so that it is displayed as several individual corners:

- In the *Corners Setup* form, select a corner group name and click *Expand corner group* on the toolbar.
- In the *Corners* section on the Setup assistant, right-click a corner group and choose *Expand Corner*.




This option 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 from 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


© 2023 Cadence Design Systems, Inc.

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.

Perform the following steps to expand corner groups for selected variables:

1. Press and hold the *Control* key and click 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

Perform one of the following steps to undo the expand of a corner group:

- In the *Corners Setup* form, right-click an expanded corner name and choose *Undo Expand Corner*.
- In the *Corners* section on the Setup assistant, right-click an expanded corner name and choose *Undo Expand Corner*.

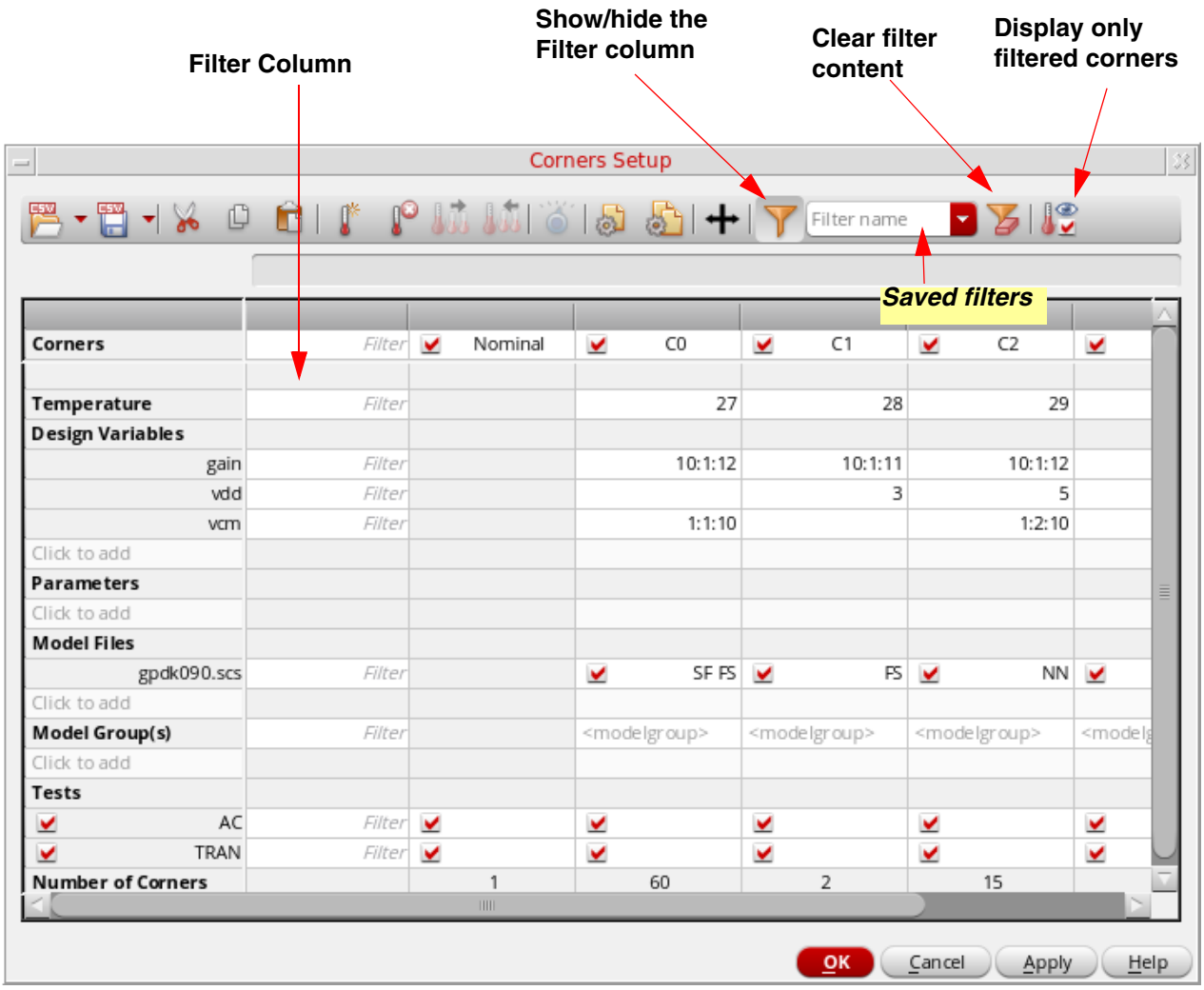
*Important*

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.


© 2023 Cadence Design Systems, Inc.

# Filtering Corners

When the number of corners is large, you can apply filters to selectively display the corners that you want to work upon.




To apply filters in the Corners Setup form:

1. Click the button in the toolbar.

   The Filter column is displayed as the first column.

2. Specify filters for the required rows.

   The accepted formats for specifying filters are:

   - Space-separated list of values, such as `SF FS NN`


© 2023 Cadence Design Systems, Inc.

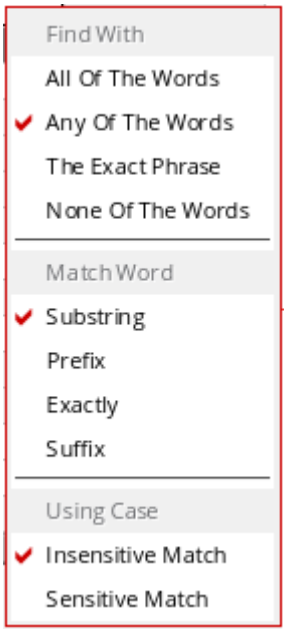
- Comma-separated list of values, such as `27,28,29`
- Range of values, such as `2:6` or `1.0:0.1:4.0`
- Enabled or disabled status in the format `enabled:yes` or `enabled:no`

  For example, if you specify `enabled:yes` for a certain row, only the enabled columns for that row will be displayed.

The corners are dynamically filtered as you type in the filter criteria.

3. (Optional) Right-click the *Filter* column to specify the following advanced filter options, as shown in the following figure.

   The options selected in this figure are the default selections.



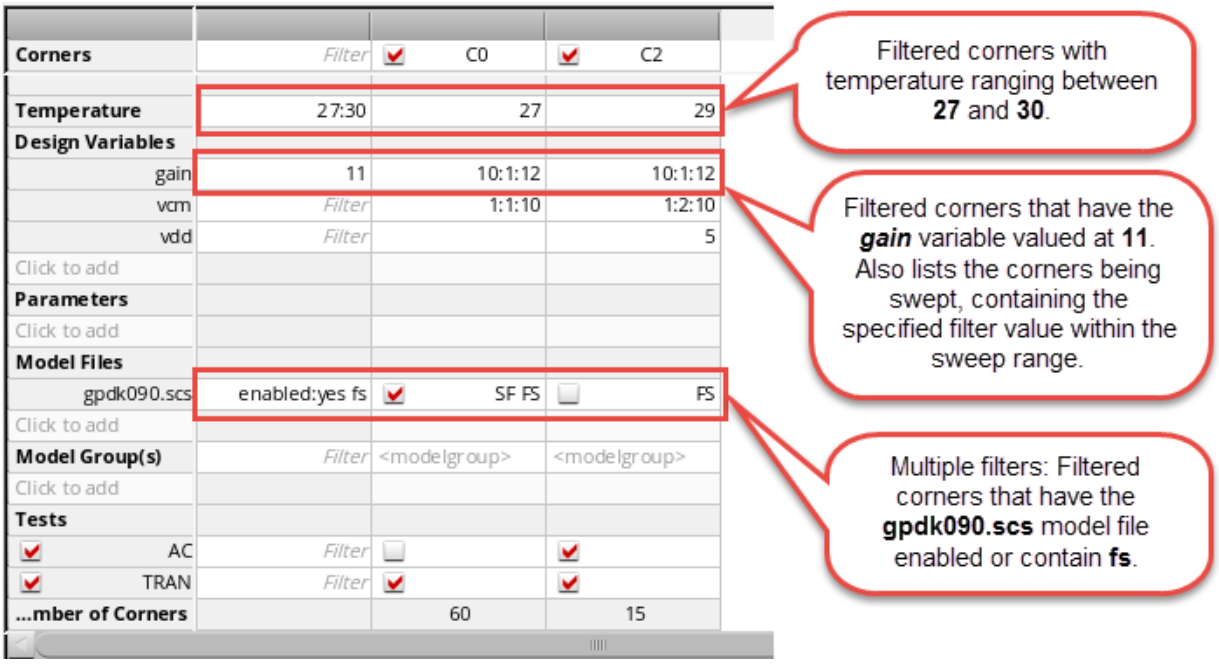


© 2023 Cadence Design Systems, Inc.

| Advanced Filter | Description |
|---|---|
| *Find With* | Using this option, you can set your filter criteria to search:<br>■ All of the words in the selected filter<br>■ Any of the words in the selected filter<br>■ The exact phrase of the selected filter<br>■ None of the words in the selected filter |
| *Match Words* | Using this option, you can set your filter criteria to match:<br>■ *Substring*: displays the columns with values containing the filter criteria<br>■ *Prefix*: displays the columns with values starting with the filter criteria<br>■ *Exactly*: displays the columns with values matching the filter criteria<br>■ *Suffix*: displays the columns with values ending with the filter criteria |
| *Using Case* | Using this option, you can specify whether the filter criteria should match casing or not. |


© 2023 Cadence Design Systems, Inc.

The following figure shows examples of three filters applied in the Corner Setup form.




In the second example, the filter value 11 matches the corners with sweep value 10:1:12 also. This is because the sweep range in expanded form is 10 11 12 and it matches the default advanced filter, *Find With: Any Of The Words*.

To clear contents of all the filters at once, click .

1. (Optional) Click the button in the toolbar to enable only the filtered corners in the simulation.

   **Note:** If you use this option to enable the filtered columns, the previously enabled corners that do not match the filter criteria will be disabled in the Data View pane.

### Multiple Filters

When you apply multiple filters to the rows with check boxes, the results are displayed based on the type of advanced filter applied to that filter. Consider the following example.

In the Corner Setup form, you want to filter the corners containing the `gpdk090.scs` model files with `FS` setting. To do so, apply a filter in the following format:

```
enabled:yes fs
```

This filter will give you results based on your selections of advanced filters.


© 2023 Cadence Design Systems, Inc.

***Case: 1***

*Find With: All Of The Words*

*Match Word: Substring*

*Using Case: Insensitive Match*

| Corners | Filter | C2 | C3 |
|---|---|---|---|
| | | | |
| **Temperature** | Filter | 29 | 30 |
| **Design Variables** | | | |
| gain | Filter | 10:1:12 | 10:2:13 |
| vcm | Filter | 1:2:10 | |
| vdd | Filter | 5 | 7 |
| Click to add | | | |
| **Parameters** | | | |
| Click to add | | | |
| **Model Files** | | | |
| gpdk090.scs | enabled:yes fs | FS | FS |
| Click to add | | | |
| **Model Group(s)** | Filter | <modelgroup> | <modelgroup> |

Gives results that match both the conditions.
So, the corners that have the `gpdk090.scs` model file enabled and also have `FS` set on them will be displayed.

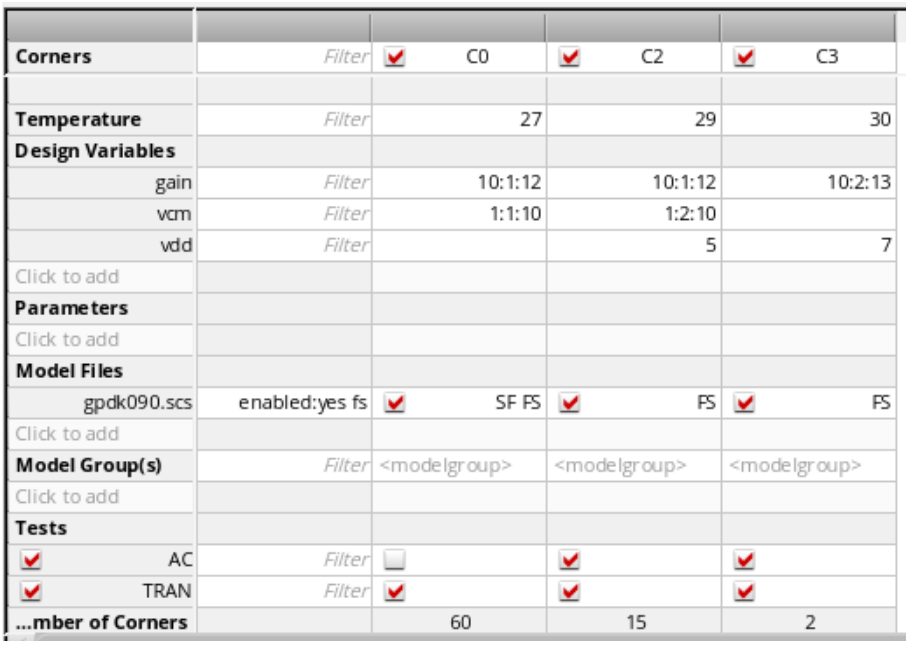
***Case: 2***

*Find With: Any Of The Words*

*Match Word: Substring*


© 2023 Cadence Design Systems, Inc.

*Using Case: Insensitive Match*



| Corners | Filter | C0 | C2 | C3 |
|---|---|---|---|---|
| | | | | |
| Temperature | Filter | 27 | 29 | 30 |
| Design Variables | | | | |
| gain | Filter | 10:1:12 | 10:1:12 | 10:2:13 |
| vcm | Filter | 1:1:10 | 1:2:10 | |
| vdd | Filter | | 5 | 7 |
| Click to add | | | | |
| Parameters | | | | |
| Click to add | | | | |
| Model Files | | | | |
| gpdk090.scs | enabled:yes fs | SF FS | FS | FS |
| Click to add | | | | |
| Model Group(s) | Filter | <modelgroup> | <modelgroup> | <modelgroup> |
| Click to add | | | | |
| Tests | | | | |
| AC | Filter | | | |
| TRAN | Filter | | | |
| ...mber of Corners | | 60 | 15 | 2 |

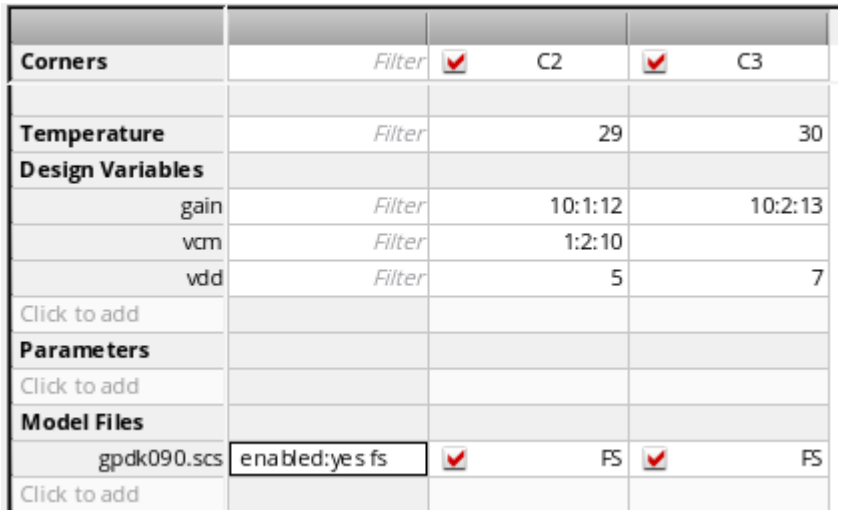
Gives results that match either of the conditions. So, the corners that have either the `gpdk090.scs` model file enabled, or have `FS` set on them will be displayed.

***Case 3:***

*Find With: The Exact Phrase*

*Match Word: Exactly*

*Using Case: Insensitive Match*



| Corners | Filter | C2 | C3 |
|---|---|---|---|
| | | | |
| Temperature | Filter | 29 | 30 |
| Design Variables | | | |
| gain | Filter | 10:1:12 | 10:2:13 |
| vcm | Filter | 1:2:10 | |
| vdd | Filter | 5 | 7 |
| Click to add | | | |
| Parameters | | | |
| Click to add | | | |
| Model Files | | | |
| gpdk090.scs | enabled:yes fs | FS | FS |
| Click to add | | | |

Gives results that match the phrase used in the filter. So, the corner that have the `gpdk090.scs` model file enabled and have only `FS` set on them will be displayed. No corners in this example.

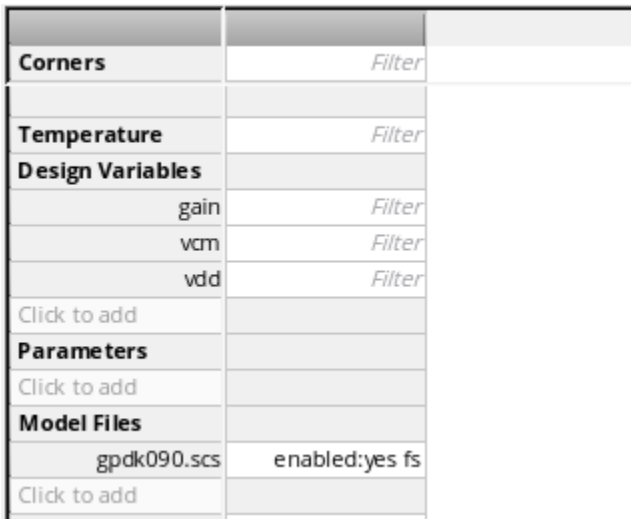

© 2023 Cadence Design Systems, Inc.

***Case 4:***

*Find With: None Of The Words*

*Match Word: Substring*

*Using Case: Insensitive Match*

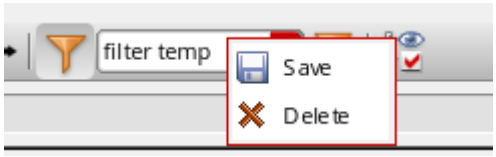



Gives results that does not match any of the conditions.
Since the specified advance filters do not match any of the rows, no corner is displayed.

## Saving Filters

You can save a set of commonly used filters through the following steps:

1. Apply filters to the rows in the Corner Setup form.
2. In the *Filter Name* field on the toolbar, enter a name for the set of filters.
3. Right-click and select *Save*. Alternatively, you can also press `Enter` to save.




The saved filter name will start appearing in the *Filter Name* drop-down list. These filters are saved in the `.sdb` and will be available for the future sessions.


© 2023 Cadence Design Systems, Inc.

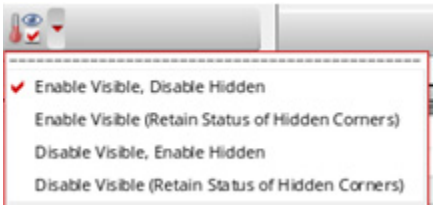
### Deleting Saved Filters

To delete a saved filter:

➡ In the *Filter Name* drop-down list, right-click the named filter that you want to delete and click *Delete*.

### Enabling or Disabling Filtered Corners

After applying filters to view the desired set of corners, you can enable or disable all the displayed or hidden corners by using the commands in the (Enable or disable filtered corners) drop-down list. The drop-down list contains the following commands:




- *Enable Visible, Disable Hidden*: Enables all visible corners and disables all hidden corners. If any hidden corner was enabled earlier, it is disabled.
- *Enable Visible (Retain Status of Hidden Corners)*: Enables all visible corners. The status of hidden corners is retained. Therefore, after this command is used, the set of enabled corners includes the visible corners and the hidden, but enabled corners.
- *Disable Visible, Enable Hidden*: Disables all visible corners and enables all hidden corners. If any hidden corner was disabled earlier, it is enabled.
- *Disable Visible (Retain Status of Hidden Corners)*: Disables all visible corners. The status of hidden corners is retained as is. Therefore, after this command is used, the set of disabled corners includes the visible corners and the hidden, but disabled corners.

## Resizing Corners

To automatically resize width of all the columns in the Corner Setup form to fit according to the corner name:

➡ In the Corner Setup form, click the (Resize Corner Columns) button.


© 2023 Cadence Design Systems, Inc.

# Simulating Corner

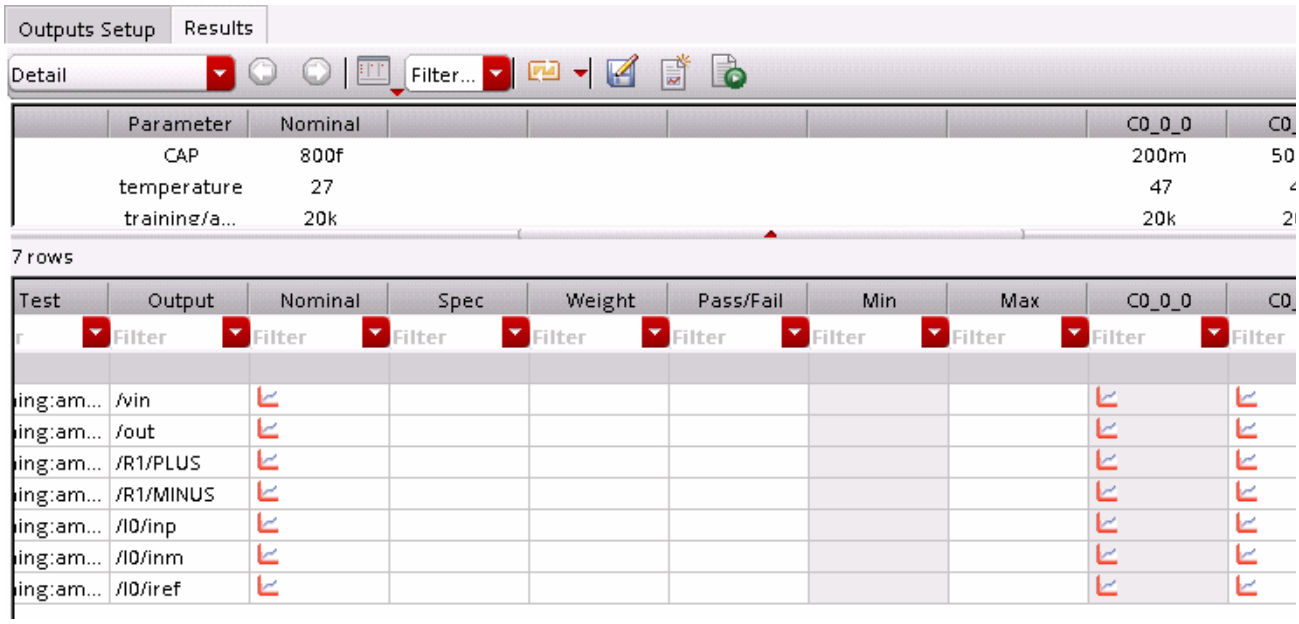
Perform the following step to run a simulation for a specific corner:

➡ In the *Corners* section on the Setup assistant, right-click a corner and choose *Run*.

Perform the following step to run a simulation for one or more corners:

1. In the Setup assistant, ensure that the name of the corners for which simulation is required to be run are selected in the *Corners* section.
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.




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_`.


© 2023 Cadence Design Systems, Inc.

*Important*

While running simulations with corners, ADE Explorer 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`.


© 2023 Cadence Design Systems, Inc.

# 7

# Running a Simulation

This chapter describes how to run a simulation that you have set up.


- Prerequisites to Run Simulation on page 598
- ADE Explorer Simulation Flow on page 599
- Setting up Job Policies on page 600
- Starting a Simulation on page 627
- Interrupting or Stopping a Simulation on page 635
- Debugging Points on page 650



© 2023 Cadence Design Systems, Inc.

# Prerequisites to Run Simulation

Before running a simulation, you need to do the following tasks:

- Start the Virtuoso® ADE Explorer and choose a design, simulator, model files, and analyses that you want to run.
- Specify the simulation setup, such as design variables, parameters
- Specify which outputs to save or plot


© 2023 Cadence Design Systems, Inc.

# ADE Explorer Simulation Flow

All simulations running in ADE Explorer are distributed according to the job control method specified in the job policy.

ADE Explorer supports the following two job control modes:

- **LSCS**: This is the default job control model.It is a highly scalable job control mode that can handle thousands of netlisting and simulation jobs in parallel. This mode contains three systems: Virtuoso, Netlist Service, and Simulation Service. These systems work independently and concurrently with the shared queues of messages. The workflow being divided into separate processes means that the resources are consumed only when they are needed. Therefore, this model enables a more efficient exchange of information between these distinct processes.

  For more details about how to configure setup and run simulation for LSCS mode, refer to Running Simulations with LSCS Job Control Mode.

- **ICRP**: This uses the IC Remote Processes that are started by ADE Assembler in the `virtuoso -nograph` mode. These remote processes perform the following tasks:

  - Generate netlists
  - Run simulations
  - Evaluate the results
  - Return the results to the ADE Assembler user interface

  Separate remote processes imply that the above tasks do not block the ADE Assembler user interface and still enable you to continue working when the simulation run is in progress.

  An ICRP process may start a few child processes (clsbd, oaFSLockD, cdsNameServer, cdsServIpc) to perform some special tasks. These child processes may become daemon and continue to exist longer than the ICRP process itself. These processes should not be killed when the original ICRP job has exited because new ICRPs that are dependent on these daemon processes may have been started in the intervening time.

  **Note:** A daemon is a program or process that runs as a background process.

- For more details about how to configure setup and run simulation for ICRP mode, refer to Running Simulations With ICRP Job Control Mode.


© 2023 Cadence Design Systems, Inc.

# Setting up Job Policies

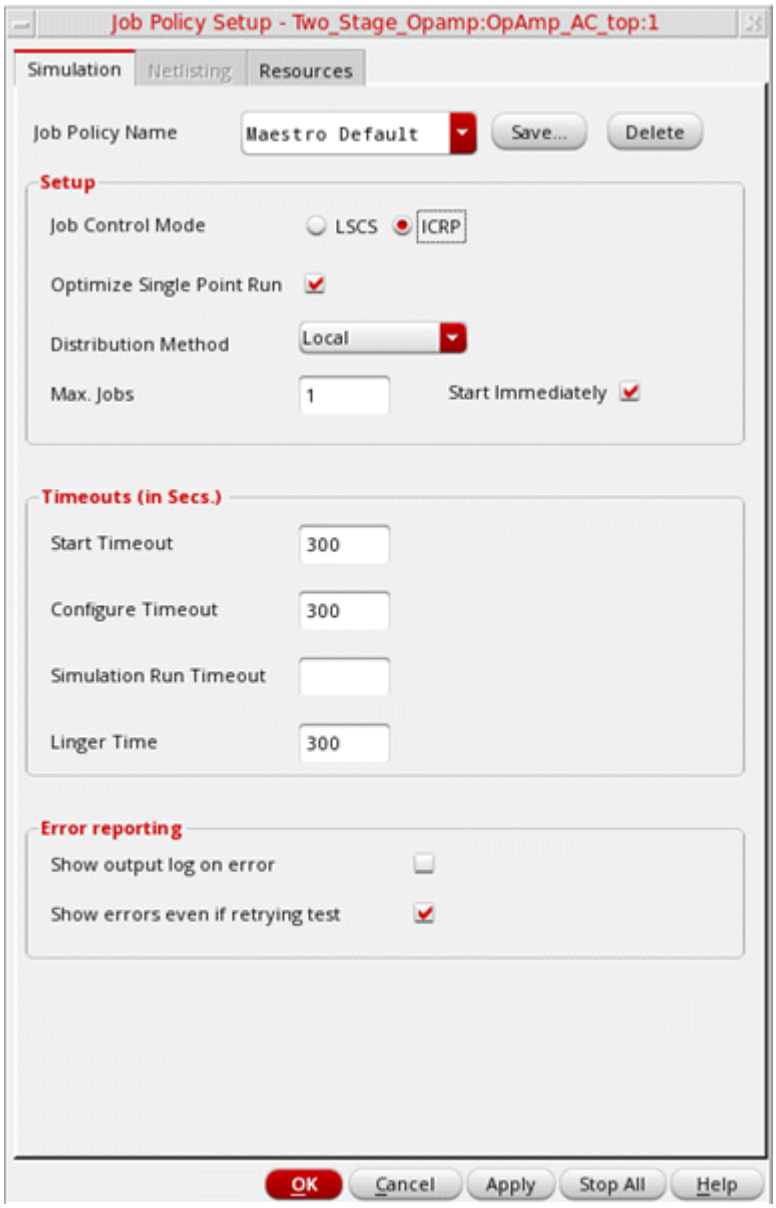
A job policy specifies the job control mode to be used to run simulations. In addition, it specifies the job distribution method: local, remote, or command; maximum number of jobs to be run at a time; and various job timeout settings.

The default job policy for ADE Explorer is `Maestro 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*.


© 2023 Cadence Design Systems, Inc.

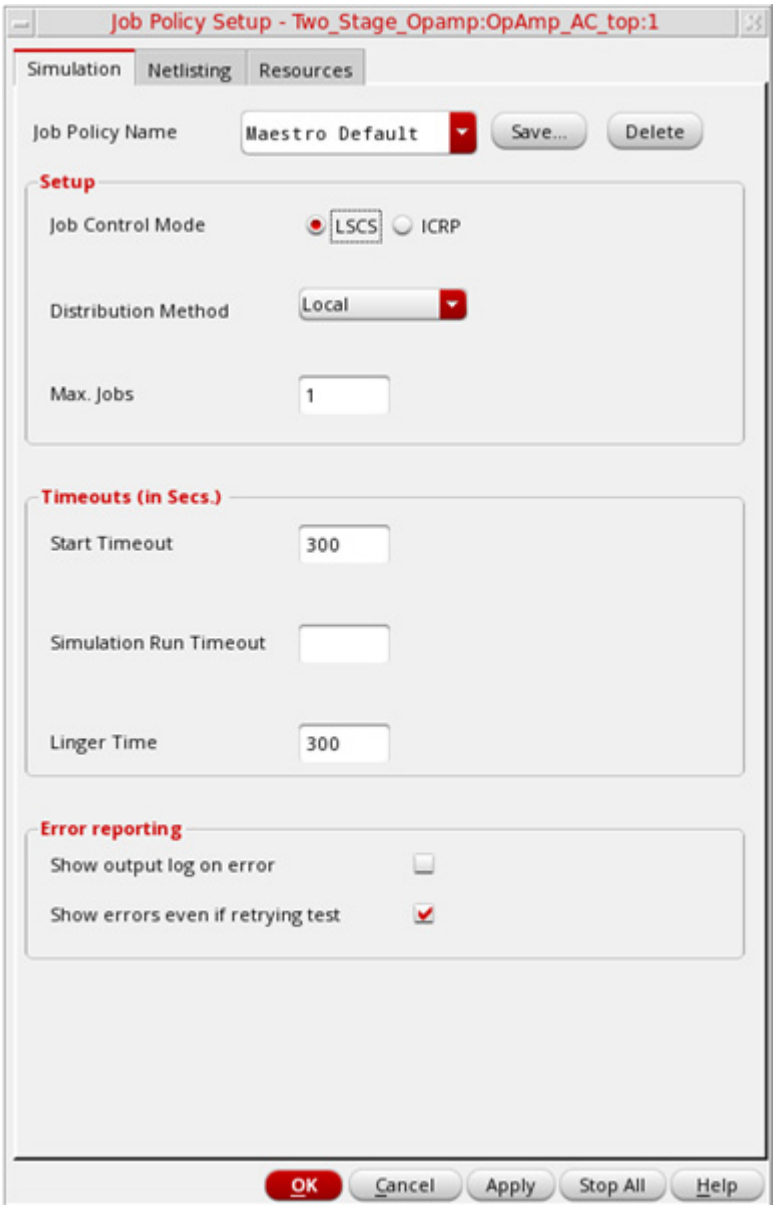
You can set up a job policy to define the distributed processing resources required to run simulations for the test.
To set up a job policy, choose *Setup – Job Setup*. The *Job Policy Setup* form appears. In this form, modify the settings to suit your requirements and save the setup with a new policy name.




To use the ICRP job control mode, click *ICRP* in the Setup section. The fields related to ICRP are displayed.


© 2023 Cadence Design Systems, Inc.

By default, the LSCS job control mode is set and the fields related to this control mode are visible.




**Note:** For the ICRP mode, the *Netlisting* tab is not available.


© 2023 Cadence Design Systems, Inc.

*Important*

The job policies and their settings are global to ADE Explorer and ADE Assembler. For example, if a change is made in a job policy or its setup in ADE Explorer for a particular test, the job policy setup will reflect in ADE Assembler as well. Similarly, if a change is made in a job policy or its setup in ADE Assembler, the change is seen in the ADE Explorer Job Policy form.

## Setting Up the Default Job Policy

To specify a customized default job policy:

1. In the Job Policy Setup Form:
   a. Specify a job policy name.

      For example, `myDefaultJobPolicy`.

   b. Change the values for the fields *Distribution Method*, *Max. Jobs*, and *Linger Time*.
2. Click the *Save* button next to the *Job Policy Name* field.

   The Save Jobpolicy form is displayed.
3. In this form, select the path for the job policy.
4. Click *OK*.
5. Set this job policy name as the value for the following variable.

   ```
   envSetVal( "adexl.icrpStartup" "defaultJobPolicy" 'string
   "myDefaultJobPolicy" )
   ```

   **Note:** By default, the job control mode is set to LSCS. Set the following variable to specify the default job policy for ICRP job control mode:

   ```
   envSetVal("maestro.distribute" "forceJobControlModeWhenViewOpen"
   'cyclic "ICRP")
   ```

If you do not specify a job policy, the program applies the following defaults.

- Distribution Method: Local
- Max. Jobs: 1
- Start Timeout: 300 (seconds)


© 2023 Cadence Design Systems, Inc.

- Linger Time: 300 (seconds)
- Show errors even if retrying test: t
- Optimize Single Point Run: t
- Communication Timeout: 180 (seconds)
- Do not estimate: t
- In the *Default Value* group box of the *Resources* tab:
  - Memory: 1000 MB
  - CPU: 1 unit(s)
- In the *Scale Estimated Values* by group box:
  - Memory: 100%
  - CPU: 100%

## Specifying a Job Policy Name

To specify a job policy name, do the following:

➡ In the Job Setup form, 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.

## Optimizing a Single Point Run

By default, the *Optimize Single Point Run* check box in the *Setup* section is selected. Therefore, for a single point run, ADE Explorer optimizes the run by completing the netlist generation and expression evaluation tasks inside the ADE Explorer process and running the simulation as per the distribution method specified by the job policy. This saves the overhead of starting a separate process thereby saving the overall processing time.

***Important points to note:***

- This feature is available only if the specified simulator is Spectre or AMS.


© 2023 Cadence Design Systems, Inc.

- While the netlist generation for a single point run is being done inside the ADE Explorer process, the user interface becomes unavailable for any changes. Therefore, it is recommended to clear the *Optimize Single Point Run* check box for designs that require a large time to generate netlist.
- This feature is not supported when:
  - Simulating a single point for the EAD flow or for Monte Carlo sampling.
  - The distribution method is set to *Command* and the command is non-interactive. An interactive command waits for the job to complete before returning control to the terminal.

## Specifying a Distribution Method

You can specify a distribution method by using the *Distribution Method* drop-down list

Depending on your setup, you can choose one of the following methods:

- `Local` – The program runs simulations on the local host.
- `Remote-Host` – The program runs simulations on the remote host that you specify.

  **Note:** On selecting *Remote-Host* as the distribution method, the *Host* field is displayed. In this field, specify the name of a valid remote host.
- `Command` – You send a command to the distributed processing software.

  On selecting, *Command* as the distribution method, the *Command* field is displayed. In this field, specify the command you want to use to start jobs.The command must adhere to these guidelines.

**Note:** Both LBS and Command distribution method submit ADE Explorer 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.

If the specified job distribution method is invalid, ADE Explorer checks whether a default distribution method is returned by the `maeGetDefaultDistributionMethod` custom SKILL function. You can define this function in the .cdsinit file as shown in the example below.

```
procedure(maeGetDefaultDistributionMethod()
"Command"
)
```


© 2023 Cadence Design Systems, Inc.

## Specifying Max Jobs

Max jobs specifies the maximum number of jobs that can run in a ADE Explorer session.

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 Explorer 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 `1000`.

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 Explorer. This means that the defined number of processes are already running before you start the simulation. This saves the overhead of staring processes at time of running simulation.

To turn off this feature, do the following:

➤ Deselect the *Start Immediately* check box.

### *Important Points to Note*

- Even if already started, the life of an idle 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 Explorer, these processes would have already been killed and would restart when a simulation is run.
- In a scenario where the resource requirements change dynamically, it is recommended not to select the *Start Immediately* check box. The initial process will start based on the previous resource requirement in the job setup, but since the resource requirement change for every new run, the previously started process may not fulfil the new requirements. Hence, the older processes need to be killed and new processes are launched on the machine that meets the resource requirement.


© 2023 Cadence Design Systems, Inc.

## Specifying an LBS Job Policy

ADE Explorer supports the following Distributed Resource Management Systems (DRMS):

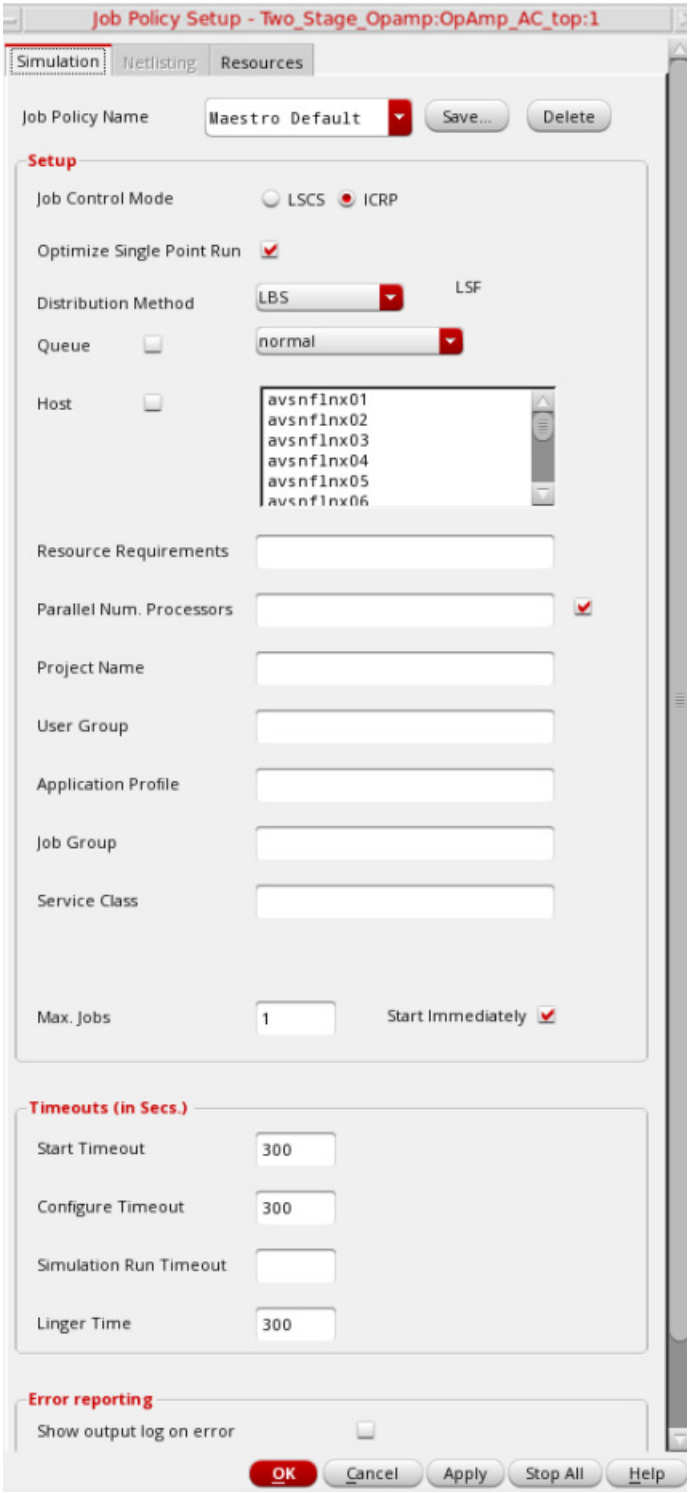
| | |
|---|---|
| LBS | Load Balancing System – Simple job distribution system (`cdsqmgr`) for setting up queues (collections of host machines) and hosts per queue; each queue has a job-per-host limit |
| LSF | Load Sharing Facility – DRMS from Platform Computing |
| SGE | Sun Grid Engine – Freeware from Sun Microsystems, Inc. |
| OpenLava | OpenLava Workload Management System |

**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.

1. Specify a name for your job policy.
2. In the *Distribution Method* drop-down list, select *LBS*.


© 2023 Cadence Design Systems, Inc.

The check boxes, fields, and list boxes related to specifying queues and hosts appear. If you have set up the LBS DRMS, the form shows the fields, as shown below:





© 2023 Cadence Design Systems, Inc.

If you have LSF in your cluster setup, the `LSF` string is displayed next to the *Distribution Method* field.

If you have OpenLava in your cluster setup, the `OpenLava` string is displayed next to the *Distribution Method* field.

If you have set up the SGE DRMS, the form shows the fields relevant to SGE.

3. (For LSF and OpenLava 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 the default system queue.

4. (For LSF and OpenLava 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, the host to be used is decided by the DRMS.

5. (For LSF and OpenLava 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 LSF and OpenLava only) In the *Parallel Num. Processors* field, specify the number of parallel processors to be used to run a simulation. To use this value, select the check box given to the right.

7. (For LSF and OpenLava only) In the *Job Group* field, specify the name of the job group to which you need to submit the job.

8. (For LSF and OpenLava only) In the *Service Class* field, specify the name of the service class to which you need to submit the job.

9. (For LSF only) Select the *Memory Host Limit* check box to apply the memory limit set for the selected LSF queue in the configuration file for LSF.


© 2023 Cadence Design Systems, Inc.

10. (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`.

11. (For SGE only) In the *Soft Resource Req* field, specify the resources that a job needs but that 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`.

12. (For SGE only) In the *Job Priority* field, specify the priority for the job being submitted relative to the other pending jobs submitted by you. The 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 the priority for a job as `-500`.

13. (For SGE only) In the *Parallel Environment* field, specify the name of the parallel environment.

14. (Optional) Specifying Max Jobs.

15. (Optional) Specifying Job Timeouts.

16. (Optional) Specifying Error Reporting Options.

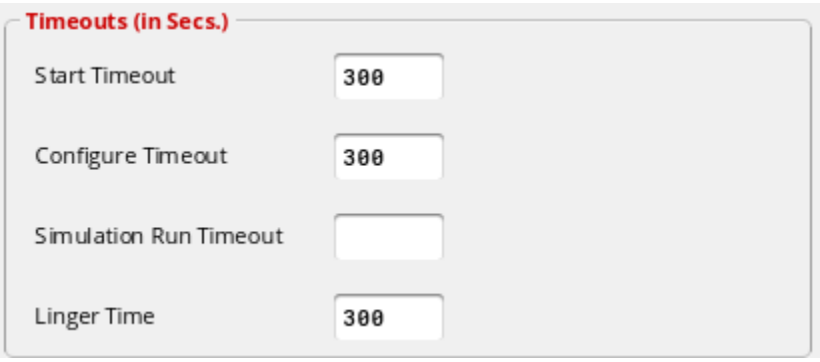
17. (Optional) Specifying Multiple Run Options.

18. Save your job policy.

    For more information, see Starting a Simulation on page 627.

19. Click *OK*.

## Specifying Job Timeouts





© 2023 Cadence Design Systems, Inc.

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 process to report back to ADE Explorer that it has started the job. The wait time starts as soon as ADE Explorer submits the job.

   If the process does not respond in the number of seconds you specify, ADE Explorer considers the job not started and kills it. If there is another job waiting to start, ADE Explorer starts it.

2. In the *Configure Timeout* field, type an integer number of seconds to wait for the process to report back to ADE Explorer that it has configured the job. The wait time starts as soon as ADE Explorer sends the job configure command.

   If the process does not respond with a successful configuration message in the number of seconds you specify, ADE Explorer considers the job configuration unsuccessful and kills it. ADE Explorer 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 process to report back to ADE Explorer that it has run the job. The wait time starts as soon as ADE Explorer sends the run command for the job.

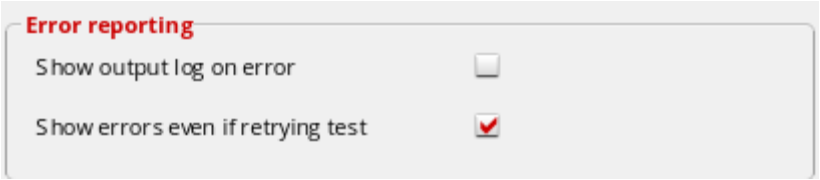
   If the process does not respond that the job has completed in the number of seconds you specify, ADE Explorer considers the job run unsuccessful and kills it. ADE Explorer looks for the next idle job and tries to start, configure, and run it.

4. In the *Linger Time* field, type an integer number of seconds after which you want the program to kill a remote job after the simulations finish.

*Important*

When a timeout value is set to `0`, `-1` or is left blank, the timeout is disabled. This leads to an unlimited wait time.

## Specifying Error Reporting Options




To specify error reporting options, do the following in the *Error Reporting* group box:


© 2023 Cadence Design Systems, Inc.

- 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.

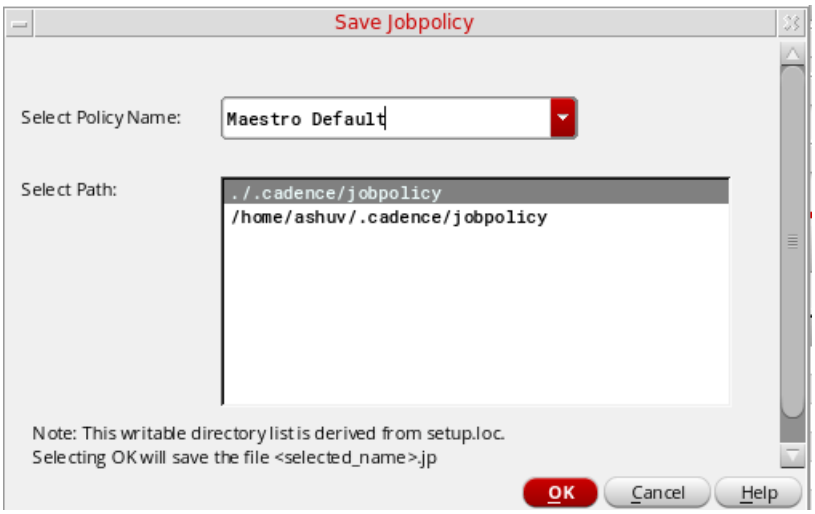
- To cause the program to display the output log file on the occurrence of an error for a test, even if the ADE Explorer distribution system is retrying the test, select the *Show errors even if retrying test* check box.

## Saving or Deleting 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` directory in the current directory
   - The `.cadence` directory in the path specified in the `CDS_WORKAREA` environment variable.
   - `$HOME/.cadence` (the `.cadence` directory in your home directory)


© 2023 Cadence Design Systems, Inc.

- The `.cadence` directory in the path specified in the `CDS_PROJECT` environment variable.
- The `.cadence` directory in the path specified in the `CDS_SITE` environment variable.

3. Click *OK*.

   The job policy is saved in the `jobpolicy` directory 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` directory in the current directory
- The `.cadence` directory in the path specified in the `CDS_WORKAREA` environment variable.
- `$HOME/.cadence` (the `.cadence` directory in your home directory)
- The `.cadence` directory in the path specified in the `CDS_PROJECT` environment variable.
- The `.cadence` directory in the path specified in the `CDS_SITE` environment variable.

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.

Click *Delete*.


© 2023 Cadence Design Systems, Inc.

# Running Simulations With ICRP Job Control Mode

The ICRP job control model uses the IC Remote Processes that are started by ADE Explorer in the `virtuoso -nograph` mode. These process in turn, may start a few child processes (clsbd, oaFSLockD, cdsNameServer, cdsServIpc) to perform some special tasks.

ADE Explorer internally uses ICRP jobs to efficiently distribute time-consuming tasks that can be performed in parallel. Settings for these jobs 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.

For more information on managing the ICRP job control mode, see the following topics:

- Specifying a Job Policy Name


© 2023 Cadence Design Systems, Inc.

# Running Simulations With ICRP Job Control Mode

The ICRP job control model uses the IC Remote Processes that are started by ADE Assembler in the `virtuoso -nograph` mode. These process in turn, may start a few child processes (clsbd, oaFSLockD, cdsNameServer, cdsServIpc) to perform some special tasks.

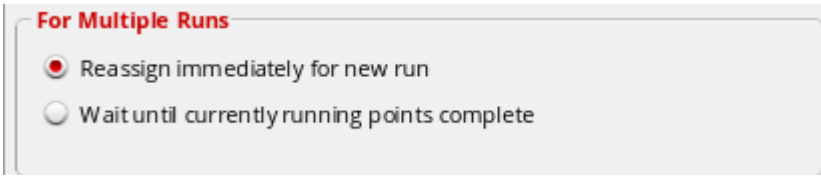
ADE Assembler internally uses ICRP jobs to efficiently distribute time-consuming tasks that can be performed in parallel. Settings for these jobs such as how many remote processes to start; where the processes should run, on local or remote machines; or the time for which a remote process should stay active and wait for a simulation to run; are set as a job policy.

For more information on managing the ICRP job control mode, see the following topics:

- Specifying a Job Policy Name
- Optimizing a Single Point Run
- Specifying a Distribution Method
- Specifying Max Jobs
- Specifying an LBS Job Policy
- Specifying Job Timeouts
- Specifying Error Reporting Options
- Saving or Deleting a Job Policy
- Specifying Multiple Run Options
- Setting Up Resource Options


© 2023 Cadence Design Systems, Inc.

## Specifying Multiple Run Options




To specify the job option to be used when running multiple runs in the same ADE Explorer 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.

## Setting Up Resource Options

ADE Explorer provides a feature to estimate the memory and CPU resources required to run simulations. After completing the simulation setup, you can use this feature to run resource estimation for the defined test, and ensure that the required memory and CPUs are available on the local or remote computer as defined by the job policy setup.

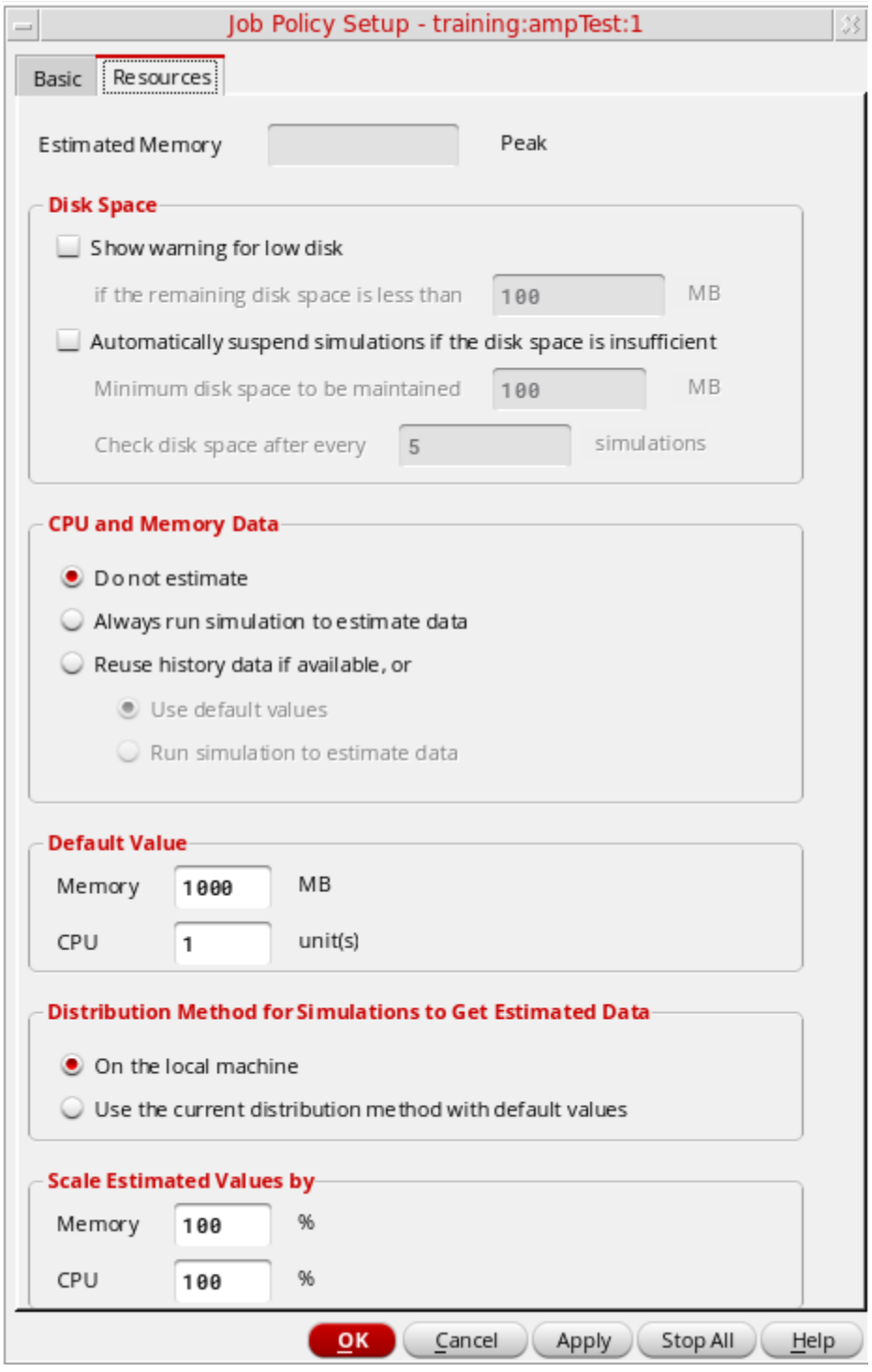
*Important*

This feature can be used only for the **hb** or **tran** analyses, or for **pss** and **qpss** when the time-domain shooting engine is used for these two analyses, that is when the *Engine* property for the **pss** and **qpss** analyses is set to `Shooting` on the Choosing Analyses form. This feature is supported by MMSIM 14.1 ISR8 or later releases.


© 2023 Cadence Design Systems, Inc.

By default, the resource estimation feature is disabled. You can enable and configure it using the *Resources* tab on the Job Policy Setup form shown below.


© 2023 Cadence Design Systems, Inc.





© 2023 Cadence Design Systems, Inc.

The following table describes the commands given on the Resources tab.

| Form Field | Description |
|---|---|
| *Estimated Memory* | Shows the estimated memory requirement to run the current simulation run. This feature is enabled using the options given in the *Provide CPU and Memory Data* group box. |
| *Provide CPU and Memory Data* | |
| *- Do not estimate* | Disables estimation of processors and memory required for the current simulation. In this case, you can yourself specify the required number of processors to be used to distribute simulations.<br><br>**Note:** When you disable estimation for LBS DRMS, select the check box given to the right of the *Parallel Num. Processors* field on the *Basic* tab of the Job Policy Setup form and specify a value in it. If you clear the check box given to the right of the *Parallel Num. Processors* field, the value given in the field is also cleared. In such a case, you should select any one of the other two options in this field to get an estimate of processors required. |
| *- Always run simulation to estimate data* | Enables the feature and runs a simulation before every new simulation to estimate resources |
| *- Reuse history data if available* | Enables the feature and estimates resources based on the data available for the most recent history available in the current ADE Explorer session.<br><br>If no history data is available, either use the default values specified in the *Default Value* group box, or runs a simulation to estimate values. |
| *Default Value* | |
| *- Memory* | Specifies the default memory requirement in Mbytes<br><br>Default value: `1000 Mbytes` |

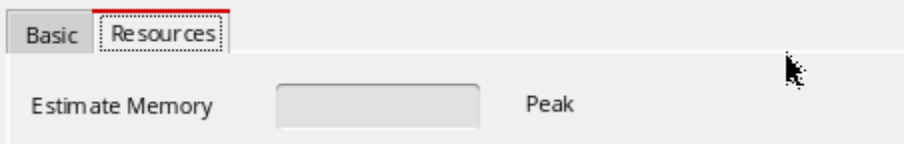

© 2023 Cadence Design Systems, Inc.

| Form Field | Description |
|---|---|
| *- CPU* | Specifies the default number of CPUs required<br>Default value: `1 processor`<br>If you know the resource requirements for the simulations you are running, you can specify the default values and select the Use Default Values option in the *Provide CPU and Memory Data* group box. |
| *Distribution Method for Simulations to Get Estimated Data* | |
| *- On the local machine* | Specifies that the simulation for resource estimation needs to be run on the local computer irrespective of the distribution methods set on the *Basic* tab.<br>This option is helpful when you cannot run simulations on a local computer. |
| *- Use the current distribution method with default values* | Specifies that the simulation for resource estimation needs to be run using the distribution method specified on the *Basic* tab.<br>This option helps in getting a good estimate of resource requirement because it uses the same remote computers that are required to run the simulations. |
| *Scale Estimated Values by* | |
| *- Memory* | Specifies the value by which the estimated value of memory requirement should be scaled<br>Default value: `100%` |
| *- CPU* | Specifies the value by which the estimated value of CPU requirement should be scaled<br>Default value: `100%` |

If this feature is enabled using the options given in the *Provide CPU and Memory Data* group box, when you run a simulation in ADE Explorer, the resource requirement is


© 2023 Cadence Design Systems, Inc.

estimated. The estimated memory required to run the simulaitons is populated in the *Estimate Memory* field on the *Resources* tab of the Job Policy setup form, as shown below.




Before the next simulation run, you can ensure that the required memory is available on the local or remote computer where you plan to run the simulation as per the Distribution Method of the job policy. You can use the template strings `%MEM_EST` and `%CPU_EST` to automatically transmit these resource requirements to your subsequent jobs.

When the distribution method is set to LBS, and you are using an LSF cluster to run simulation, the value of the required number of processors is displayed in the *Parallel Num Processors* field on the *Basic* tab.This value is used with other resource requirement settings for LSF. An example of a resource requirement string for LBS is shown below.

```
rusage[mem=%MEM_EST:mmsim=4] span[hosts=1]
```

When the distribution method is set to *Command*, you can include the `%CPU_EST` and `%MEM_EST` template string in the command, as shown below.

```
bsub -n %CPU_EST -I -q queueName -R \"rusage[mem=%MEM_EST]\" other_bsub_options
```

During runtime, the `%CPU_EST` and `%MEM_EST` template strings are replaced with the estimated values

# Suspending and Resuming Simulation Runs

A resource-intensive simulation run might fail when sufficient disk space is not available. In such cases, you can suspend one or all of the running simulations, clean up the disk, and resume those for completion. You do not need to cancel the run.

You can suspend a simulation in the following ways:

- Automatically Suspend Simulations When the Disk Space is Low
- Manually Suspend Simulations

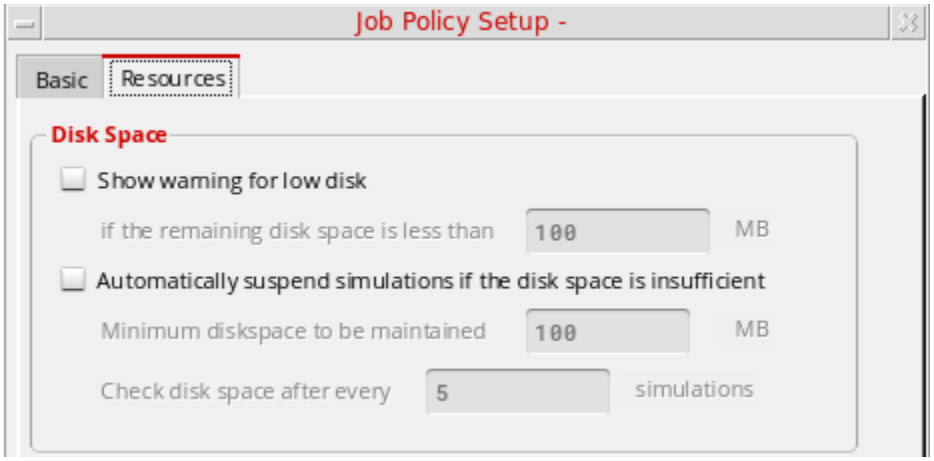
## Automatically Suspend Simulations When the Disk Space is Low

You can configure the commands to automatically suspend simulations when the disk space is low, and flag appropriate information or warning messages.


© 2023 Cadence Design Systems, Inc.

To enable checking of disk space and automatic suspension of simulations depending on the configured settings, do this:

1. Choose *Options – Job Setup* to open the Job Policy Setup form.
2. Open the Resources tab, shown below.




3. Specify the following options in the *Disk Space* group:
   - *Show warning for low disk*: Select this check box to enable checking of the available disk space when a simulation run is started. ADE Explorer shows a warning in these cases:
     - If the available disk space before running simulation for the first point is less than the threshold value specified in the *If the remaining disk space is less than _ MB* field, a warning message is flagged to report about the unavailability of the required space and to confirm whether the run is to be continued.
     - For the Single Run, Sweeps, or Corners run mode, after the first design point is simulated, the disk space consumed for that point is used to estimate the total space required to run simulations for all the points. If the **remaining (available – estimated)** disk space falls below the threshold, ADE Explorer shows the following types of messages:

       • **If the remaining (available – estimated) disk space is more than or equal to the threshold** specified in the *If the remaining disk space is less than* field, an information message is flagged to report the same.

       • **If the remaining (available – estimated) disk space is less than or equal to the threshold** specified in the *If the remaining disk space is less than* field, a warning message is flagged to report the same. It is recommended to manually suspend the simulation run for the current history, free up the disk space, and resume the run.


© 2023 Cadence Design Systems, Inc.

Consider an example of a simulation run for three design points.
- The simulation of the first design point used: 15 MB
- The estimated disk space for the remaining two points: 30 MB
- The available disk space after the first design point is simulated: 115 MB
- The threshold value for disk space: 100 MB
- The estimated remaining disk space = 115 – 30 = 85 MB, which is less than the threshold value

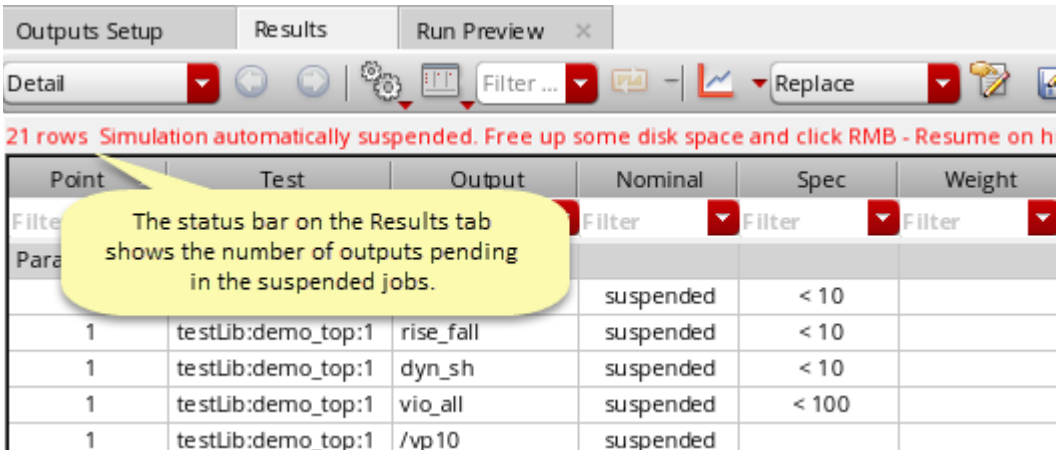
In this scenario, after the simulation for the first design point is complete, a warning message is displayed to clear disk space.

- *Automatically suspend simulations if the disk space is insufficient*: Select this check box to enable automatic suspension of all the simulations in progress whenever the available disk space falls below the minimum threshold specified in the *Minimum disk space to be maintained (in MB)* field.

  When automatic suspension is enabled, by default, ADE Explorer checks the disk space after every five simulations. You can change the frequency in the *Check disk space after every _ simulations* field.

4. Click *OK* to close the form.

   ADE Explorer uses the settings specified on this form to constantly monitor the available and estimated disk space and shows appropriate messages. If a simulation is automatically suspended, the status on the Run Summary tab and the Results tab is updated, as shown below.



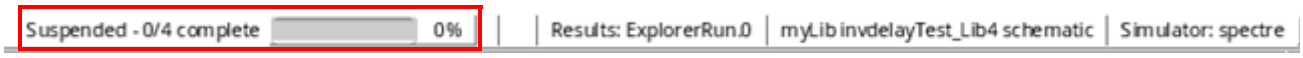

In the Explorer Run Summary pane, the icons of the jobs allocated for the suspended simulation run appear as paused ( ). Their tooltips also show the time since when they are in the suspended state.


© 2023 Cadence Design Systems, Inc.

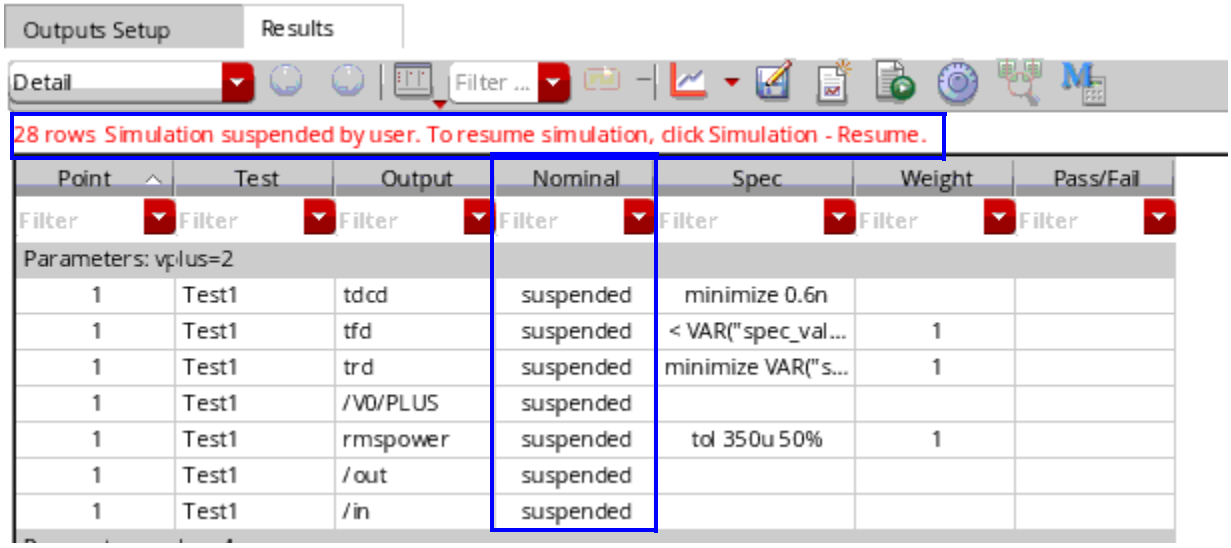
## Manually Suspend Simulations

You can manually suspend the simulation while a simulation run is in progress. To do so, choose *Simulation – Suspend*.

The current simulation run is suspended and the following status is displayed on the ADE Explorer status bar to indicate that the simulation run is suspended.




To indicate that the simulation is manually suspended for a point, the status of the suspended point on *Results* tab shows the text, `Suspended`, as shown in the figure below. In addition, the message highlighted in figure below is displayed in the table header to indicate that the current simulation run has been manually suspended.




To resume a suspended simulation, choose *Simulation – Resume*. The simulation is resumed from the point where it was suspended previously.

### *Important Points to Note*

- The settings for automatic suspension of jobs can be set only in the global Job Policy Setup form. These settings are not available in the job policy at the test level.
- You cannot suspend a simulation when the continueICRPRunOnAbruptGUIExit environment variable is set to `t`.


© 2023 Cadence Design Systems, Inc.

- When you suspend a simulation, the status of its points on the *Results* tab appears as `suspended`.
- The job policy settings to automatically suspend simulations are ignored while running simulations using OCEAN or SKILL scripts.
- Starting from MMSIM 15.1 ISR15, Spectre supports the `+disk_check=<threshold value>` command-line option to automatically suspend simulations if the disk space is below the given threshold limit. To use this feature of Spectre, specify this option in the *User Command-line Options* field in the Environment Options form in ADE Explorer. When this option is given and Spectre suspends a simulation, the corresponding ICRP is suspended. If the resources running other the jobs submitted by other ICRPs have enough resources, they continue to run. For more information about Environment Options form, see Setting up Simulator Environment Options.

  When the Spectre sessions for all the ICRP jobs in ADE Explorer are suspended, the status of the history tab is changed to suspended and its background color is changed to yellow.

### Suspending or Resuming Simulations From a DRMS

When the simulations are running in a DRMS, if you need to preempt any simulation to manage the resources, send the SIGTSTP signal to the ICRP process. The signal is then sent to the Spectre process spawned by that ICRP to suspend the simulation and release the license. To retrieve the license and resume simulation, send the SIGCONT signal to that ICRP process.

When suspended, Spectre will release the license only if it is called with the `+lsuspend` option. If `+lsuspend` is not used, Spectre will suspend the simulation, but the license will not be released.

### Troubleshooting jobs that stay pending until Timeout

Certain jobs may stay pending until timeout because a Message Passing System (MPS) communication cannot be established between the ICRP and GUI processes.

Any one of the following reasons can cause this gap in communication:

- There may be a firewall blocking the TCP/IP communication between machines.
- ADE may have started a large number of jobs and does not have the sufficient time frame to respond back to all jobs. This causes the communication timeout to fail.

  **Note:** Communication timeout is a subset of the Start Timeout, and refers to the time taken to establish a TCP/IP communication.


© 2023 Cadence Design Systems, Inc.

- The cdsNameServer versions may be incompatible. If the machine initially ran a 5141 (or a version prior to 6.1.3) cdsNameServer then any attempt to establish connection using a cdsNameServer version later than 6.1.3 would fail.

  **Note:** The cdsNameServer program is used to establish communications between Cadence programs. It relies upon the TCP port number, 7325. In addition, whenever needed, the cdsNameServer program starts automatically and exits itself after being idle. To know more, see *Cadence Application Infrastructure User Guide*.

***Important points to note about MPS:***

- When virtuoso is running on the GUI_Host, a cdsNameServer process should be running on the GUI_Host listening to port 7325. This can be ensured by:

  ```
  GUI_Host% mpsinfo -c
  ```

- This listener port must also be accessible from the Remote_host. This can be confirmed by:

  ```
  Remote_Host% mpsinfo GUI_Host -c
  ```

- Access to port 7325 from remote hosts is necessary, but insufficient for the ADE flow. There are other ports which are allocated from the ephemeral range on the GUI_Host that must be accessible from remote hosts.

- Ephemeral ports are generally determined by the OS. For example, ephemeral ports have the range from 32768 to 61000 for Linux.

These ports (7325 and ephemeral ports) need to be opened up for both the GUI host and the remote host, in case these are blocked by a firewall.


© 2023 Cadence Design Systems, Inc.

# Starting a Simulation

After you specified the required settings, perform the following steps to start a simulation:

- Choose *Simulation – Netlist and Run*.

  The netlist will reflect any design changes. Any edits done in the design or the properties form would reflect in the new netlist. You should use this option when you run the simulation for the first time and also when you have edited your design. Using this option ensures that your design, the ADE setup and the output netlist for simulation are synchronized.

- Choose *Simulation – Run*.

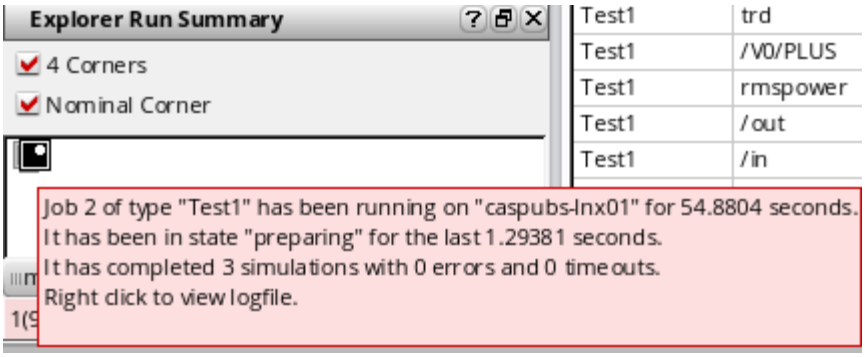
- Click the button on the Run toolbar.

  Using these options, the design is not netlisted if a netlist is already available. This option is faster than the *Simulation – Netlist and Run* option, and is useful when no design changes have been made. The resulting simulation reflects simulation setup modifications such as analysis setup changes, design variable changes, and simulator option changes.

  When a simulation is running, the status bar at the bottom of the ADE Explorer window shows the different stages of simulation run. The right-most corner of the progress bar appears on the bottom right of the ADE Explorer window displays the results directory, which is `ExplorerRun.0` by default and the library, cell, and view information.

  When the simulation is multi-point, a progress bar '`Running now`' on each row of the nominal column is displayed along with the progress bar at the bottom. When the progress bar at the bottom shows `1/4`, it indicates that one point is run out of total 4 design points.

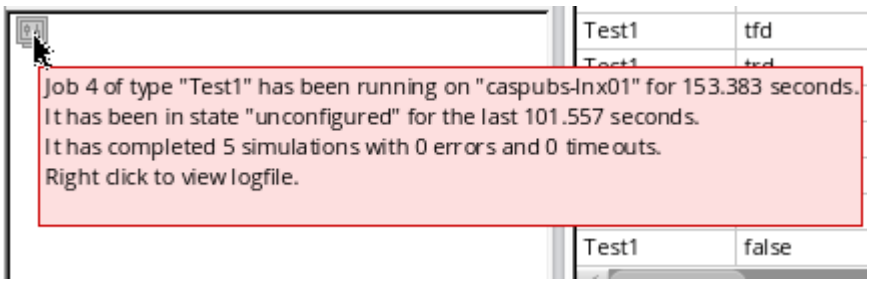
  A progress symbol ( ) appears on the Explorer Run Summary assistant. When you mouse hover this symbol, the simulation run status appears in a tooltip, as shown in the figure below:



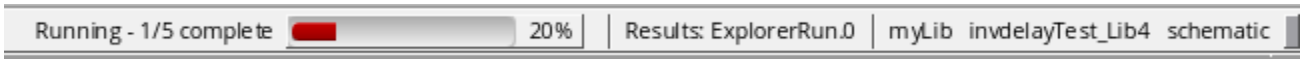


© 2023 Cadence Design Systems, Inc.

When the simulation run is complete, it shows the completion status, as shown in the figure below:




In addition, the simulation status is displayed in the status bar of the ADE Explorer window, as shown in the figure below:




When you modify the design in Schematic editor and run the simulation without saving the design updates, a warning message appears that informs about the unsaved design changes and asks whether you want to continue with the simulation run.


© 2023 Cadence Design Systems, Inc.

## Simulation RunTime Improvements

The following enhancements show significant in simulation runtime.

### Performance Enhancements Using Spectre Plugin

To reduce the performance overhead of creating the complete netlist for every simulation point and the associated Spectre startup time, a Spectre plugin has been introduced in this release. ADE Explorer creates an initial netlist for the first design point and sends it to Spectre. Spectre reuses this netlist for all the design points and makes only variable changes for which the plugin sends the details to Spectre.

If netlist recreation and Spectre startup time is comparable to simulation time, then significant performance improvement can be noticed when the new Spectre plugin is used. However, the performance improvement may diminish when the simulation run time is high.

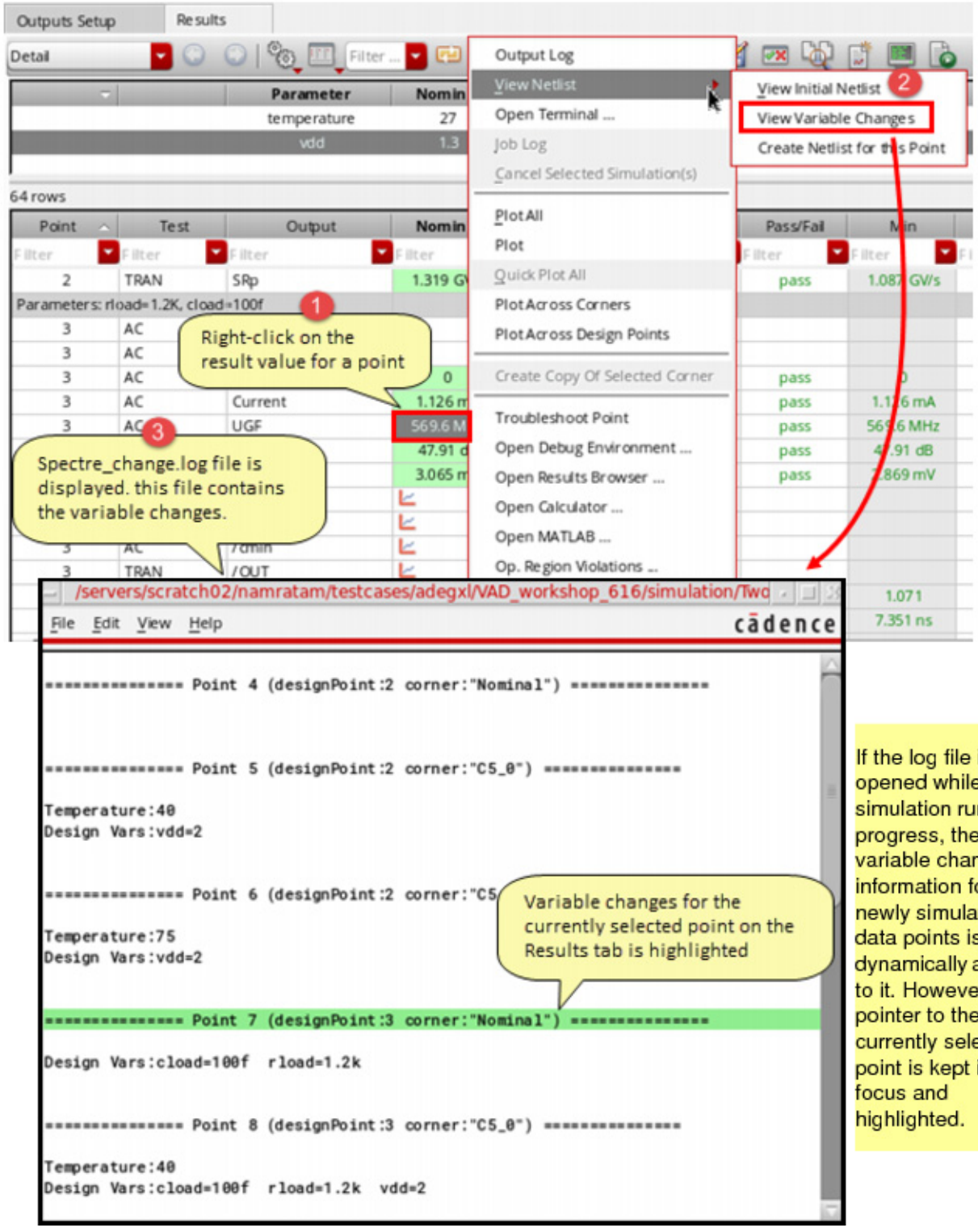
#### *Viewing Netlist*

To view or create the netlist for a design point during the simulation run, right-click in a cell in the results table and select one of the options in the *View Netlist* menu:

- *View Initial Netlist*—Opens the initial nelist generated when the first design point is simulated.
- *View Variable Changes*—Opens the `spectre_change.log` file that displays the variables changed for the current point with respect to the previous point. This file also show a log of which variables are used at which point.


© 2023 Cadence Design Systems, Inc.

- *Create Netlist for this Point*—Creates netlist for the selected point only.




If the log file is opened while the simulation run is in progress, the variable change information for the newly simulated data points is dynamically added to it. However, the pointer to the currently selected point is kept in focus and highlighted.

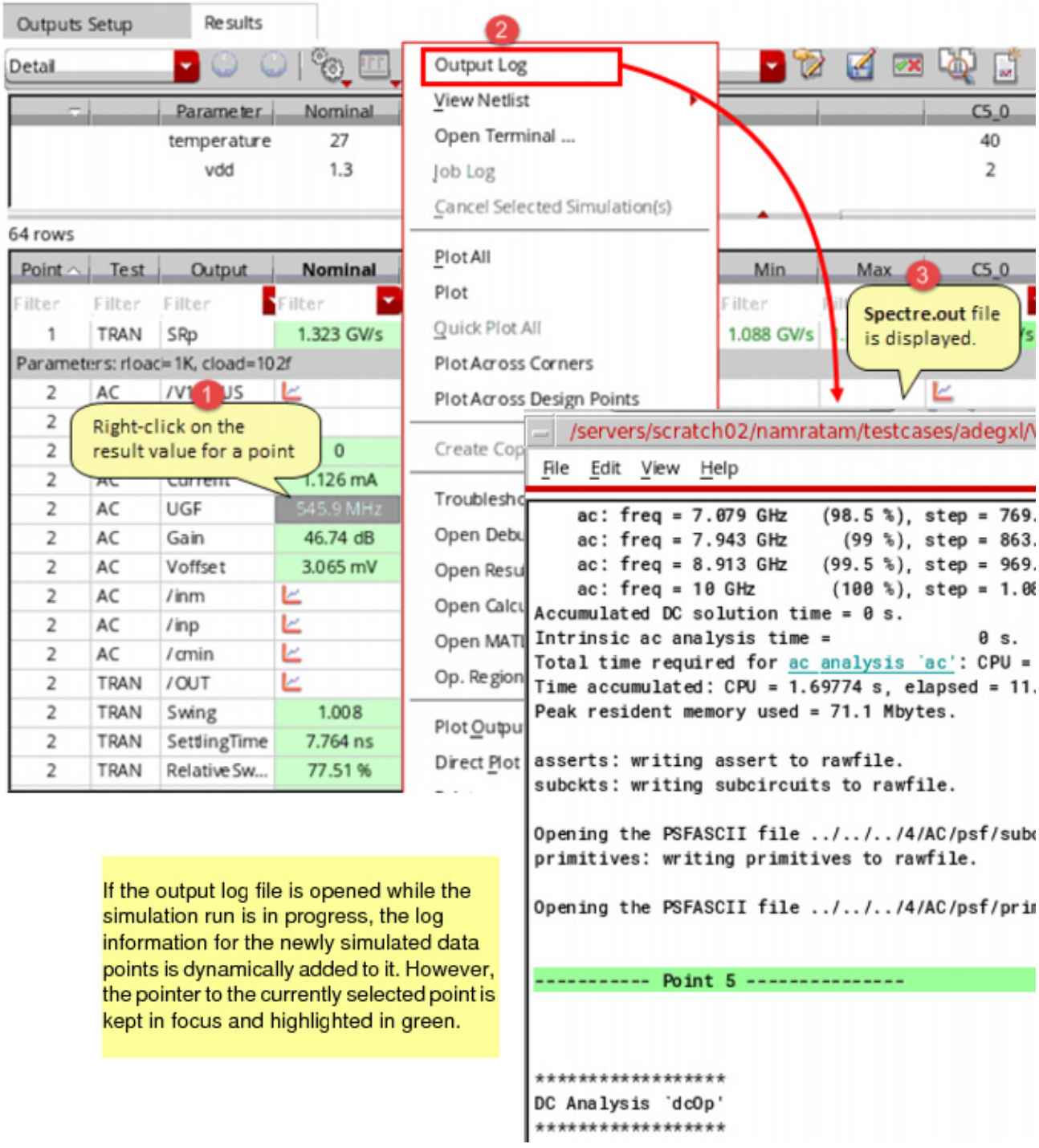

© 2023 Cadence Design Systems, Inc.

### *Viewing Output Log*

Spectre plugin maintains a single output log named `Spectre.out` for all the points. To view the output log for a data point, right-click in a cell on the *Results* tab and choose *Output Log*.


© 2023 Cadence Design Systems, Inc.

**Note:** Clicking the hyperlinks for `.s\\d+p` and `.snp` files in the `Spectre.log` file opens the S-parameter data in the Results Browser workspace in ViVA XL. You can then choose the waveforms you want to plot.




If the output log file is opened while the simulation run is in progress, the log information for the newly simulated data points is dynamically added to it. However, the pointer to the currently selected point is kept in focus and highlighted in green.


© 2023 Cadence Design Systems, Inc.

### *Limitations*

The Spectre plugin is not supported and is automatically switched off for the following features:

- Multi-technology Simulations
- Checks and Asserts
- Simulators other than Spectre
- Monte Carlo
- Statistical corner
- Troubleshoot run
- Design variables of type string
- Device parameters of type string
- Include analysis
- +xps command line option is set
- Spice outputs
- Model file sections using VAR()
- EM/IR
- DSPF files used in the parasitic aware design flow

**Note:** Spectre plugin cannot run simulations successfully if the enabled corners use the model sections, but the default model file is not defined and the Nominal corner is disabled. In this case, ADE Explorer automatically disables Spectre plugin, and continues to run simulation.

## Performance Enhancements by Simulating Group of Points

Prior to this release, ADE Explorer used to send one point at a time to ICRPs. To improve the performance of simulation throughput and the responsiveness of the user interface, ADE Explorer now sends a group of points to each ICRP.

Group run reduces the overhead of multiple interactions of the user interface with the ICRPs. In addition, when combined with the redistribution feature that automatically reallocates simulations waiting on slow resources to other resources, it helps improve the overall runtime.


© 2023 Cadence Design Systems, Inc.

To enable redistribution, set the minGroupSizeSplitAcrossIdleJobs `.cdsenv` variable to the count after which redistribution should take place. By default, this variable is set to 0.

For example, when you set this variable to 20, if an ICRP has 20 or more simulations and meanwhile there are some idle ICRPs, they are redistributed to other available ICRPs.

Consider another example where all ICRPs have completed their simulations, but 5 simulations are still pending on one ICRP, which is a slow resource. In this case:

- Redistribution helps if ADE Explorer takes away those 5 simulations and assigns them to 5 faster resources.
- Redistribution does not help if one or more of the resources receiving these new simulations are equally slow.
- Redistribution helps to a great extent if each simulation takes a few hours, and the slow resource with 5 simulations has become the bottleneck for the entire run to complete, and, there are other resources that are available.

However, the amount of improvement depends on the following factors:

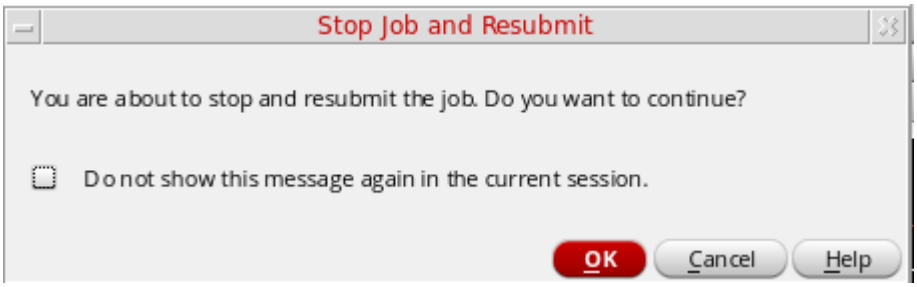
- Number of points being run
- Long simulations (if simulation runtime is large, then other improvements may appear non-significant)
- Overall quality of the farm
- Amount of load on farm machines
- Number of max jobs


© 2023 Cadence Design Systems, Inc.

# Interrupting or Stopping a Simulation

To stop a running simulation,

- Choose *Simulation – Stop*.
- On the Run toolbar, click the stop button ( ).
- Right-click the progress icon on the Explorer Run Summary assistant and click *Stop and Resubmit*. The following message appears:




The simulation is stopped and system saves any simulation results that are calculated. The stopped simulation cannot be continued.

*Important*

Clicking *Simulation – Stop* option does not only stop a running simulation, it also releases the *Spectre* license.

You can then modify the settings in the Setup and restart the simulation.

### *Important Point to Note*

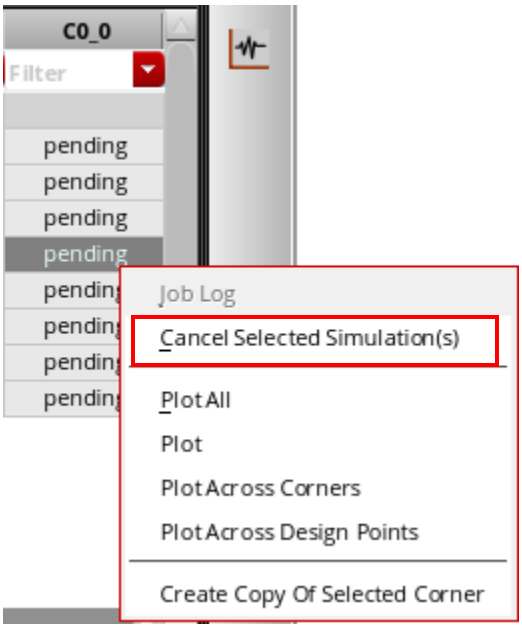
- Instead of stopping the entire simulation run, you can stop the simulation for only selected tests or points. For more details, refer to Canceling Simulations for Selected Corners on page 635.

## Canceling Simulations for Selected Corners

To cancel a simulation for one or more corners, do the following:


© 2023 Cadence Design Systems, Inc.

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.

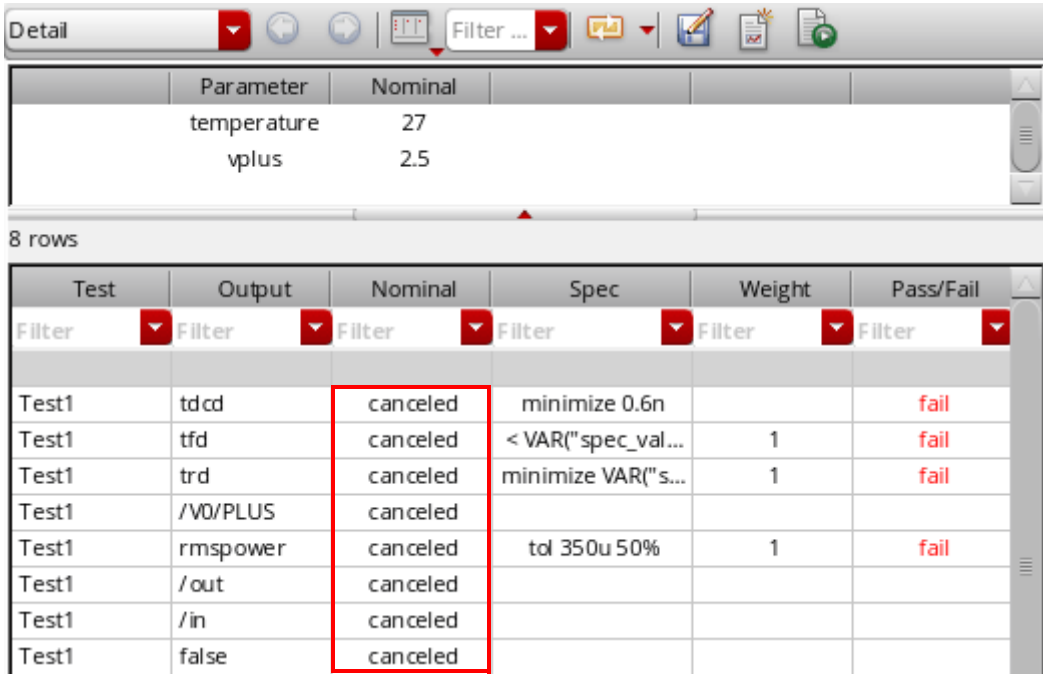



**Note:** You can cancel the simulation in `pending` as well as `running` state.

2. Right-click one of the selected points and choose *Cancel Selected Simulation(s)*.


© 2023 Cadence Design Systems, Inc.

ADE Explorer 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.

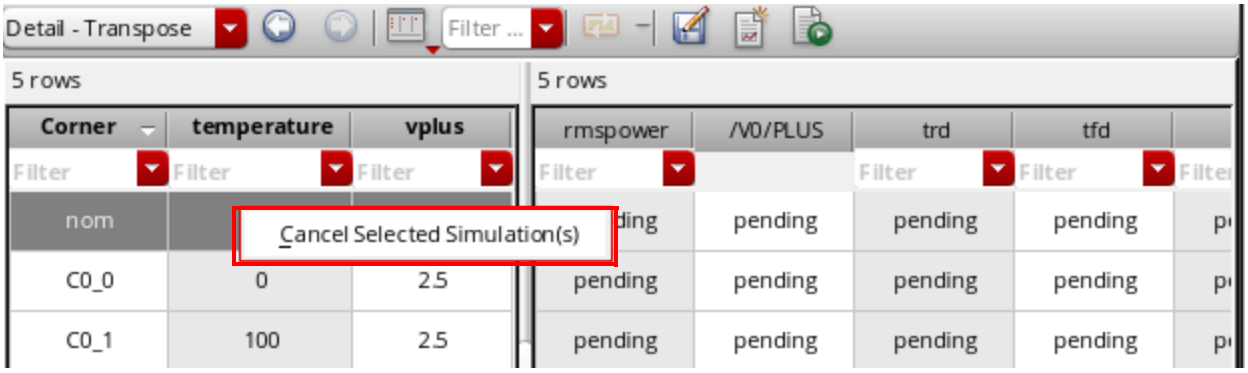



### *Important Points to Note*

- 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.
- 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:


© 2023 Cadence Design Systems, Inc.

- 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)*.




Simulations for the corresponding design point and corner combinations are canceled across all the tests.

- To cancel simulations for specific design point, and corner, select the corresponding cells in the right pane of this view, right-click and choose *Cancel Selected Simulations(s).*


© 2023 Cadence Design Systems, Inc.

## Viewing Run Preview

The *Run Preview* tab provides a summary of all the settings applied in the current ADE Explorer view. You can view various details, such as the total number of sweep points and corners that are run, the name of the test, or the job policy being used. It also displays the combinations of sweeps and corners that will be run. You can view a particular data point in the debug environment and run a simulation for that point. In addition, you can create netlist for a particular point. On this tab, you can also select specific points for which you want to run simulations. By default, this tab is not displayed.

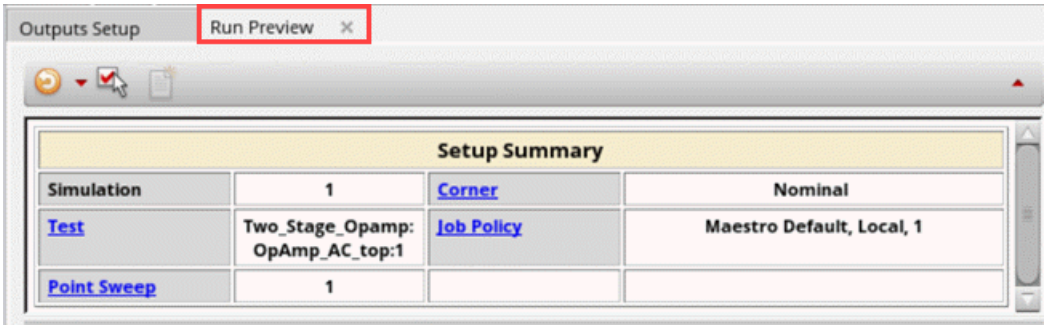
Video

For more details, you can also view the Run Point Selection video demonstration.

To open the *Run Preview* tab,

- Choose *Simulation – Run Preview.*
- Right-click anywhere on the tab bar and choose *Run Preview*.

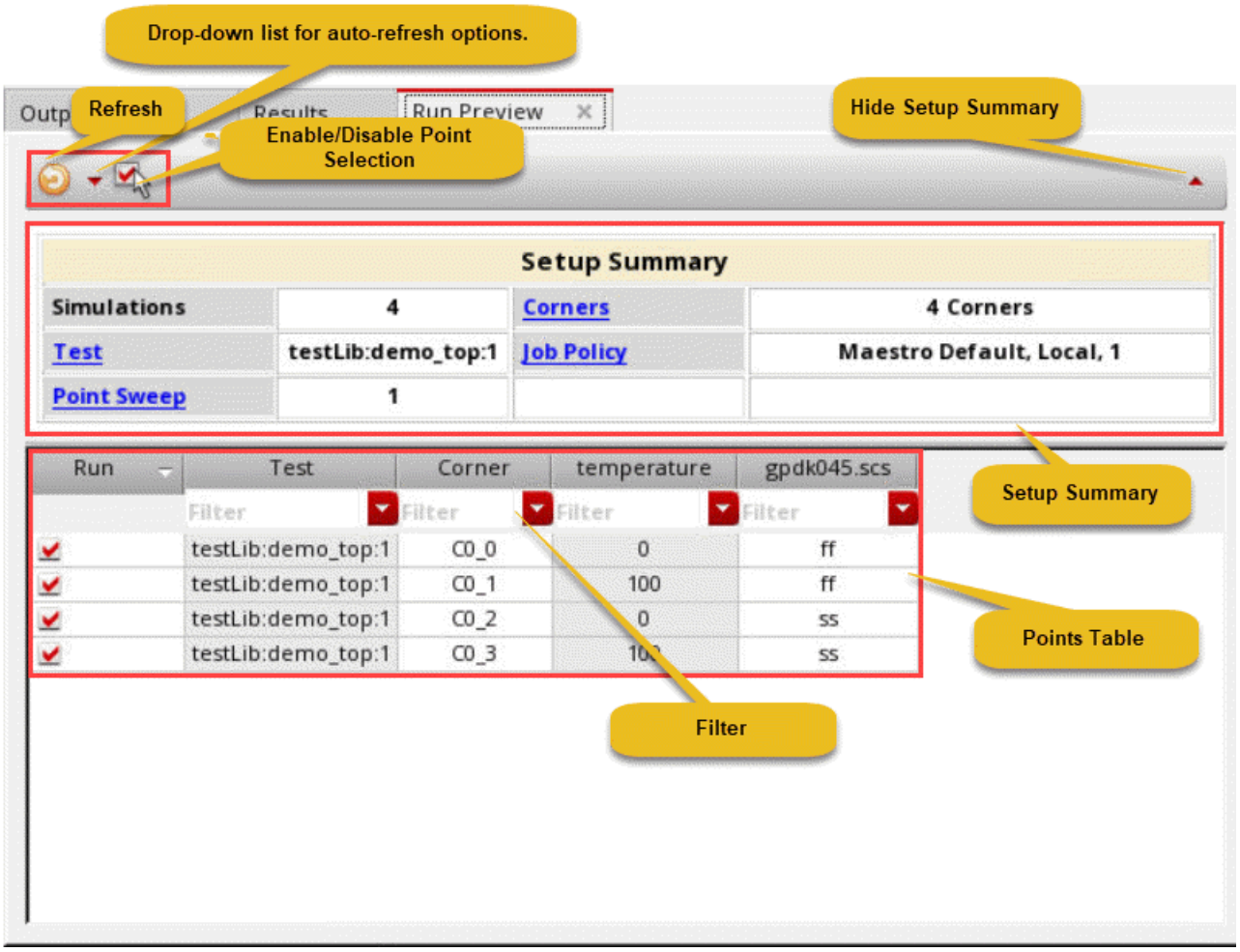
The default display setting of this tab is controlled by the `outputTabsShowDefault` environment variable.




The *Run Preview* tab has two sections—*Setup Summary* and points table, as shown in the figure below. The design point and corner combinations, and the values of variables are


© 2023 Cadence Design Systems, Inc.

displayed in the points table. You can also enable or disable run points in the points table. For more information, see Enabling or Disabling Run Points in Points Table.



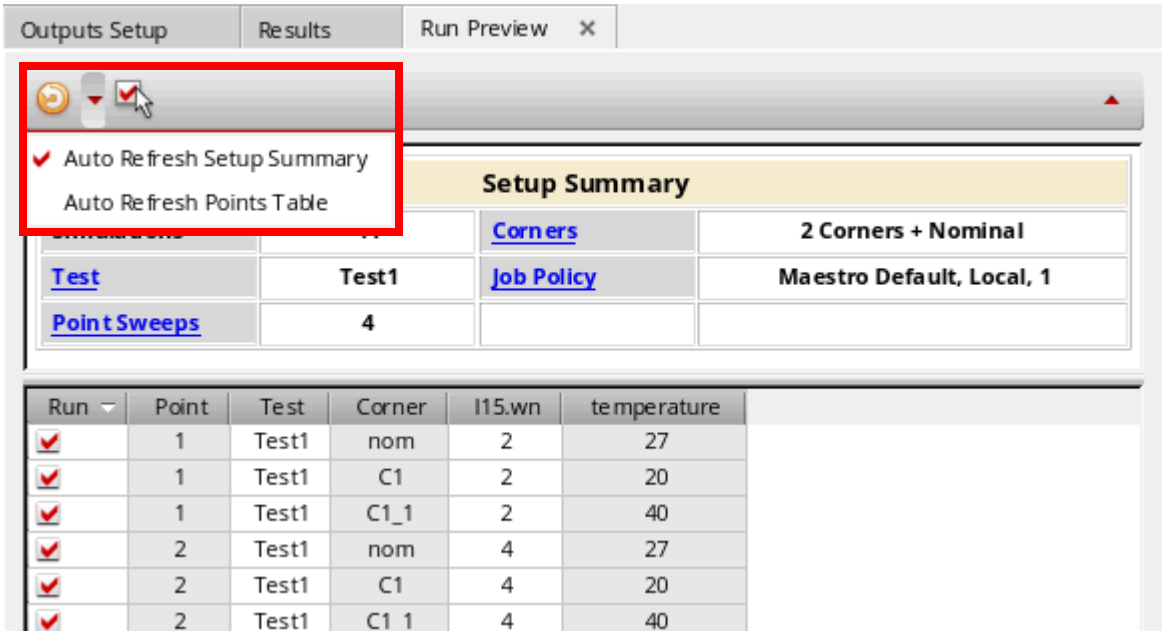

You can click *Disable Point Selection* on the upper-right corner to show or hide the *Setup Summary* section. To find a specific point or set of points in the points table, you can sort the data displayed in each column by clicking the arrow button available on the column headers. If required, you can also view specific data points by using the filters available in each column header.

## Refreshing the Setup Summary and Points Table

As you modify the setup in the current view, the setup summary is refreshed automatically and the count of corners, points, and total simulations to be run keeps varying. By default, only the summary data is displayed and updated. This is because the *Auto Refresh Setup*


© 2023 Cadence Design Systems, Inc.

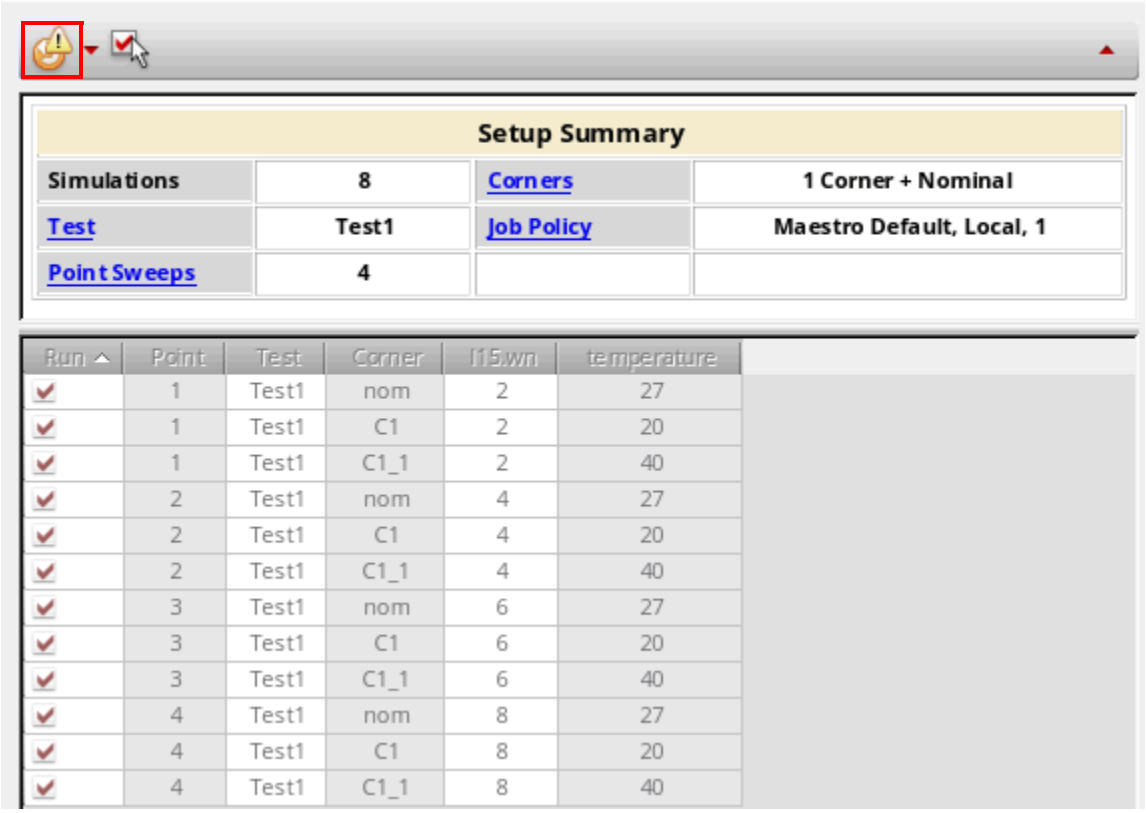
*Summary* check box in the drop-down menu is selected by default. You can click the *Refresh* button to display the details of the run points in points table.




When you open a new setup or move from ADE Explorer to ADE Assembler and vice-versa, the points table is not displayed by default. You need to click the *Refresh* button to view the points table. In addition, the points table is not automatically refreshed and becomes unavailable when the setup changes. In this case, an alert icon is displayed on the *Refresh*


© 2023 Cadence Design Systems, Inc.

button to indicate that the table data is not current. You can click the Refresh button to update the table contents.



**Setup Summary**

| | | | |
|---|---|---|---|
| Simulations | 8 | Corners | 1 Corner + Nominal |
| Test | Test1 | Job Policy | Maestro Default, Local, 1 |
| Point Sweeps | 4 | | |

| Run | Point | Test | Corner | l15.wn | temperature |
|---|---|---|---|---|---|
| ✓ | 1 | Test1 | nom | 2 | 27 |
| ✓ | 1 | Test1 | C1 | 2 | 20 |
| ✓ | 1 | Test1 | C1_1 | 2 | 40 |
| ✓ | 2 | Test1 | nom | 4 | 27 |
| ✓ | 2 | Test1 | C1 | 4 | 20 |
| ✓ | 2 | Test1 | C1_1 | 4 | 40 |
| ✓ | 3 | Test1 | nom | 6 | 27 |
| ✓ | 3 | Test1 | C1 | 6 | 20 |
| ✓ | 3 | Test1 | C1_1 | 6 | 40 |
| ✓ | 4 | Test1 | nom | 8 | 27 |
| ✓ | 4 | Test1 | C1 | 8 | 20 |
| ✓ | 4 | Test1 | C1_1 | 8 | 40 |

To refresh the points table automatically after every change in the sweep values, corner, or any other setup changes:

- Set the `enableAutoRefreshPointsTable` .cdsenv variable to `t`.
- Click the *Refresh* drop-down list and select the *Auto Refresh Points Table* option.

If the number of the simulation points is large, it may take several minutes to display the details.

*Important*

If the *Auto Refresh Points Table* check box is selected, the simulation run triggers refreshing the points table. If the number of the simulation points is large, it may take more time for the simulation run to complete. Therefore, it is recommended to clear the *Auto Refresh Points Table* check box before running simulations.

To refresh only the points table and not the Setup Summary, you can clear the *Auto Refresh Setup Summary* check box and select the *Auto Refresh Points Table* check box.


© 2023 Cadence Design Systems, Inc.

***Related Environment Variables***

- `enableAutoRefreshSetupSummary`
- `enableAutoRefreshPointsTable`

## Enabling or Disabling Run Points in Points Table

The points table shows a *Run* column that includes a check box against each row. By default, the check box is selected for all the points. The *Run* column is used to select or deselect run points for the simulation run.

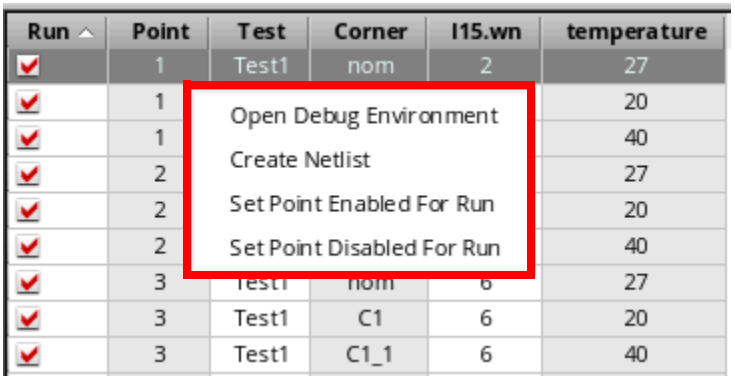
If you do not want to run a specific point, you can do one of the following to disable the point:

- Clear the check box in the *Run* column
- Right-click the point and choose *Set Point Disabled For Run*. You can choose *Set Point Enable For Run* to enable a specific point.

  The selected point is disabled.

To enable or disable a set of points:

➡ Hold down the `Ctrl` key and select the points, right-click and choose *Set Point Enable For Run* or *Set Point Disabled For Run*, respectively.




### Enabling or Disabling Run Column

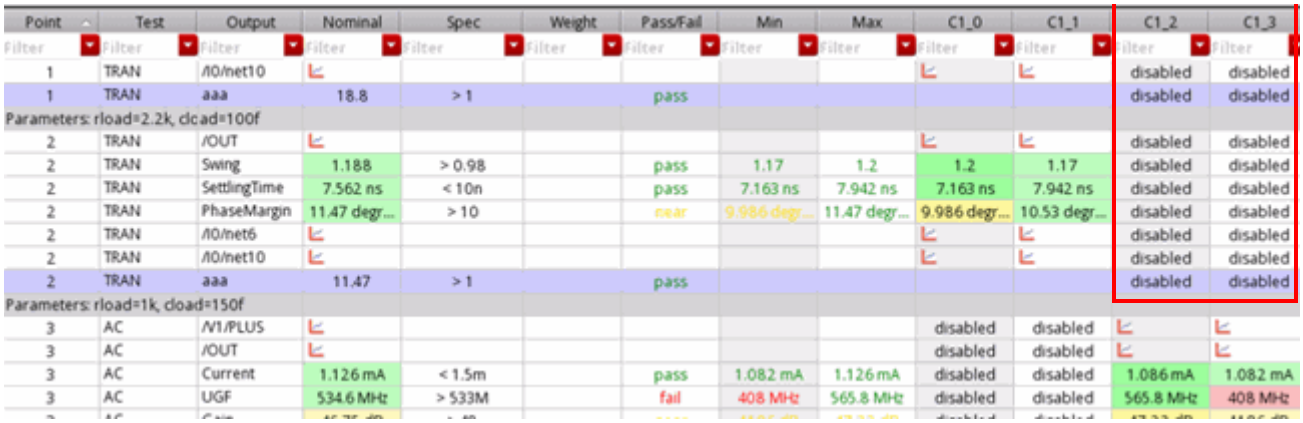
To completely disable the *Run* column and run point selection,

➡ Click the button.

The *Run* column becomes unavailable. In this case, all the points will be run irrespective of the selection you made previously. To enable the *Run* column, click this button again.


© 2023 Cadence Design Systems, Inc.

**Note:** When you disable corners or sweeps, the results in `Detail` and `Detail-Transpose` views display `disabled` in the disabled sweep-corner cell, as shown in the figure below.



| Point | Test | Output | Nominal | Spec | Weight | Pass/Fail | Min | Max | C1_0 | C1_1 | C1_2 | C1_3 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | TRAN | /I0/net10 | | | | | | | | | disabled | disabled |
| 1 | TRAN | aaa | 18.8 | > 1 | | pass | | | | | disabled | disabled |
| Parameters: rload=2.2k, cload=100f | | | | | | | | | | | | |
| 2 | TRAN | /OUT | | | | | | | | | disabled | disabled |
| 2 | TRAN | Swing | 1.188 | > 0.98 | | pass | 1.17 | 1.2 | 1.2 | 1.17 | disabled | disabled |
| 2 | TRAN | SettlingTime | 7.562 ns | < 10n | | pass | 7.163 ns | 7.942 ns | 7.163 ns | 7.942 ns | disabled | disabled |
| 2 | TRAN | PhaseMargin | 11.47 degr... | > 10 | | near | 9.986 degr... | 11.47 degr... | 9.986 degr... | 10.53 degr... | disabled | disabled |
| 2 | TRAN | /I0/net6 | | | | | | | | | disabled | disabled |
| 2 | TRAN | /I0/net10 | | | | | | | | | disabled | disabled |
| 2 | TRAN | aaa | 11.47 | > 1 | | pass | | | | | disabled | disabled |
| Parameters: rload=1k, cload=150f | | | | | | | | | | | | |
| 3 | AC | /V1/PLUS | | | | | | | disabled | disabled | | |
| 3 | AC | /OUT | | | | | | | disabled | disabled | | |
| 3 | AC | Current | 1.126 mA | < 1.5m | | pass | 1.082 mA | 1.126 mA | disabled | disabled | 1.086 mA | 1.082 mA |
| 3 | AC | UGF | 534.6 MHz | > 533M | | fail | 408 MHz | 565.8 MHz | disabled | disabled | 565.8 MHz | 408 MHz |

The completely disabled rows and columns are not displayed in the results.

***Some Important Points to Note:***

- The run point selection is not displayed when Reliability analysis is enabled. In addiiton, it currently works for only single run sweeps and corners.
- Measurements across corners, sweeps, all will ignore the disabled run points.
- By default, the columns for fixed parameters are not displayed in the points table on Run Preview tab. However, if the fixed parameters are enabled in the *Results* tab for the `Detail - Transpose` view, they will be displayed in the points table as well. To display columns for fixed parameters on *Results* tab, you can choose the *Fixed Parameters* option in the *Configure what is shown in the table* drop-down list available on the toolbar.
- The run point selection is reset and the points table is disabled when you transition between ADE Explorer to ADE Assembler and vice-versa.
- The run point selection is not considered in Saved Ocean XL scripts.
- Any setup changes in the variables, parameters or corners will reset the check boxes in Run column and select all of them.
- The run point selection setup is persistent and is saved and loaded along with view.

## Using the Information Displayed on the Run Preview Tab

You can use the information displayed on the *Run Preview* tab, as explained below.


© 2023 Cadence Design Systems, Inc.

### Viewing the Corner and Sweep Combinations

If you have selected the check boxes that enable automatic refresh of information, the details on the *Run Preview* tab show the updated corner and point details while you modify the ADE Explorer setup.

### Modifying the Setup

You can click a hyperlink provided in the *Setup Summary* section to open the relevant section in the ADE Explorer setup and modify the settings. For example, when you click the *Corner* hyperlink, the Corner Setup form is displayed where you can modify the corners for the current setup. After you close the Corners Setup form, the number of corners is updated according to the changes.

Similarly, you can click the *Job Policy* hyperlink to open the Job Policy form to view or modify the job policy. If the test has an overridden job policy, the Job Policy information in the Setup Summary indicates that by using a keyword `(various)`.

### Creating Netlist

You can create a netlist for a specific design point and corner combination and verify the details. To do this, right-click anywhere in the row corresponding to that combination and choose *Create Netlist*. ADE Explorer creates and displays a netlist for that point in a separate window.

### Running a Simulation for a Specific Point

Before running a simulation for all the points, you might be interested in viewing the results of a specific design point and corner combination. In that case, you can run a simulation for only that combination and verify results. If required, make further modifications in the setup. To run a simulation for a specific design point and corner combination, right-click anywhere in the row corresponding to that combination and choose *Open Debug Environment*. ADE Explorer opens that simulation point in the ADE Debug Simulation Environment where you can run a simulation and view the results.

**Note:** The *Create Netlist* and *Open Debug Environment* commands are not enabled for the OCEAN-based tests.


© 2023 Cadence Design Systems, Inc.

### Modifying the Format of the Run Preview Tab

The format of information displayed on the *Run Preview* tab is defined by the `previewSetup.xsl` stylesheet saved at the following default location:

`<Virtuoso-installation-directory>/share/cdssetup/adexl/previewSetup.xsl`

To modify this format or to define the content to be displayed on this tab, you can provide the path to a customized `.xsl` stylesheet by setting the `RUNPREVIEW` shell environment variable before running Virtuoso, as shown below.

```
setenv RUNPREVIEW_XSL <path-to-cutomized-stylesheet-file.xsl>
```

ADE Explorer uses the stylesheet specified by this variable to display the information on the *Run Preview* tab.


© 2023 Cadence Design Systems, Inc.

# Setting the Debugging Options

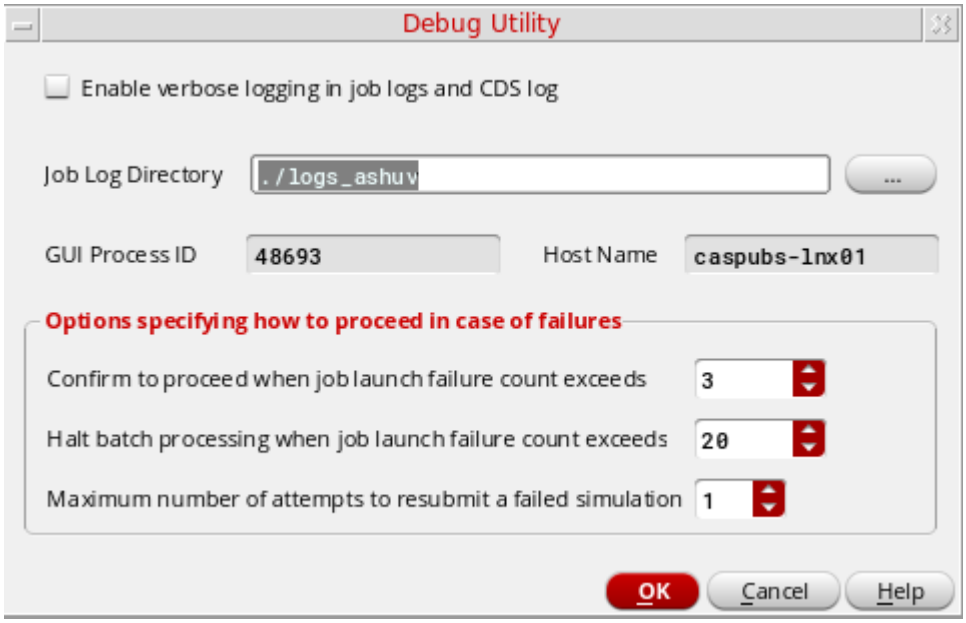
**Note:** Read a blog related to this at **Virtuosity: Let's Have Fun with ADE Debugging – Part 1**.

You can set debugging options that help you troubleshoot several problems that you face while working with ADE Explorer, such as non-responsive user-interface, ICRPs not starting, simulations showing error or warning messages, and so on. When set, these debug options are instantly applied on all the active ICRPs, however, the current simulation run remains unaffected.

Perform the following steps to set the debugging options:

1. Choose *Tools – Debug*.

   The Debug Utility form appears.




2. In this form, specify the following fields to set the debugging options:

   a. *Enable verbose logging in job logs and CDS log*—Select this check box to enable the debugging options listed below:

      - Write the debug information and prepend a time-stamp in each debug message
      - Write the callback evaluation information
      - Capture the full stack trace if an error is thrown
      - Write the Monte Carlo debug information


© 2023 Cadence Design Systems, Inc.

- Write the calcVal debug information
- Write the debug information for wave spec evaluation or re-evaluation
- Print the DE integration debug information and log all SKILL_CALLS issued by the UI
- Report information during runObjFile and interactive plotting operations
- Write the OCEAN debug information

When you click this check box, the following environment variables are enabled to write the std out and std error messages to the CDS.log file:

```
"adexl.icrpStartup" "showJobStdout" 'boolean t

"adexl.icrpStartup" "showJobStderr" 'boolean t
```

**b.** *Job Log Directory*—Specify a location where you want to redirect the job logs. This field is useful when you select the *Enable verbose logging in job logs and CDS log* field and want to write the job log files to a different location so that they can be shared for further debugging.

Alternatively, you can set the location using the following environment variable:

```
“adexl.distribute” “jobFileDir” ‘string “<myDir>”\
```

**c.** *GUI Process ID*—Show the GUI process ID.

**d.** *Host Name*—Show the host name of the machine on which you are running Virtuoso.

**e.** *Options specifying how to proceed in case of failures*—Set the following fields to specify how you want to proceed when failures are found:

- *Confirm to proceed when job launch failure counts exceeds*—Specify a count after which if a job launch is failed, an error message is displayed.

  For example, if you specify this field value as `3`, then if the job fails three times, no error message is displayed. However, if the job continues to fail for the fourth time, an error message is displayed in CIW. This indicates that the number of attempts permitted to retry a job launch in the given case is three.

  Alternatively, you can set the following environment variable to specify how many number of times to retry a job launch before reporting the error:

  ```
  "adexl.distribute" "maxJobFailPerPolicy" 'int 1000
  ```

- *Halt batch processing when job log failure count exceeds*—Specify a count after which if a job has failed to launch, the batch processing is stopped.


© 2023 Cadence Design Systems, Inc.

The value you specify is applied to all the scripts generated from the current session.

Alternatively, you can set the following environment variable to specify how many number of times a job launch can be retried before stopping the batch processing of jobs:

```
"adexl.distribute" "maxJobFailPerPolicyInBatch" 'int 1000
```

- *Maximum number of attempts to resubmit a failed simulation*—Specify how many times ADE Explorer can try resubmitting a failed simulation. By default, ADE Explorer resubmits a failed simulation only once, and if the simulation fails again, an error message is displayed.

  Alternatively, you can set the following environment variable to specify the maximum number of attempts to resubmit a failed simulation:

  ```
  envSetVal("adexl.distribute" "numRetriesOnError" 'int 2)
  ```

**f.** Click *OK* to apply the specified settings.

### Important Points to Note

- The debug options settings specified in this form will be applied on all the jobs (ICRP sessions) that are new or waiting to run a simulation. To apply these changes on the currently running jobs as well, restart the jobs by right-clicking the ICRP icon in the Run Summary assistant and choosing *Stop and Resubmit*.
- The default values displayed for the debug options in this form are the values specified for them in the related environment variables.
- The debug options settings specified in this form temporarily override the specified `.cdsenv` or `.cdsinit` settings. The form settings are applicable only to the current Virtuoso and ICRP sessions, and form fields will show the default values in the next sessions.


© 2023 Cadence Design Systems, Inc.

# Debugging Points

After you run simulations in ADE Explorer, 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 Explorer Debug Environment.

You can use the ADE Explorer Debug environment to load the setup of a data point you want to debug. The entire setup for the data point selected in ADE Explorer 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 Explorer such as running simulations and plotting results, except the following tasks:

- 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.

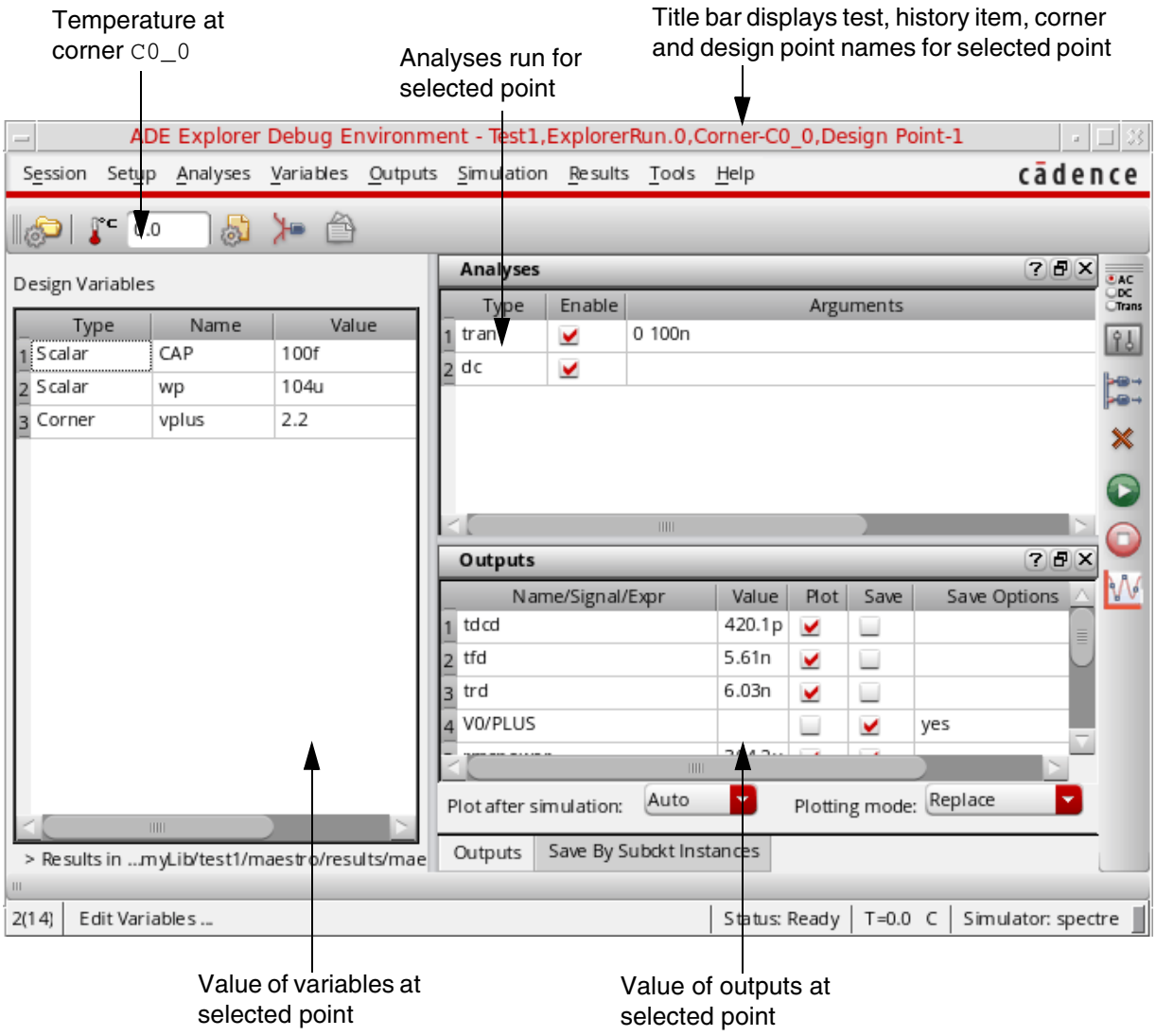
After debugging the data point, you can bring back the corrected simulation setup to the ADE Explorer view. You can then run the simulation by using the updated setup in the ADE Explorer view.

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*.


© 2023 Cadence Design Systems, Inc.

The *ADE Explorer Debug Environment* window appears displaying the setup for the data point. For example, the following figure displays the setup for corner `C0_1` of test `Test1` at data point `1` in history item `ExplorerRun.0`.




Note the following:

- The ADE Explorer 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 the above figure displays the point information as:

  `Test1, ExplorerRun.0, Corner-C0_0, Design Point-1`

  Where, `Test1` is the test name, `ExplorerRun.0` is the history item name, `C0_1` is the corner name and `3` is the design point number for the selected point.


© 2023 Cadence Design Systems, Inc.

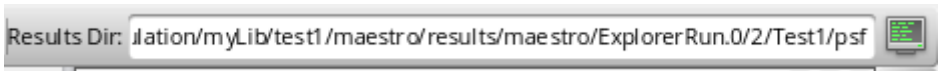
- The *Design Variables* pane displays the value of the design variables at the selected data point. The *Type* column indicates the type of the variables, as described below:

| 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<br><br>**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<br><br>**Note:** The value of the *Temperature* variable specified for the corner is displayed on the toolbar, as shown below: |

- The default project directory for the data point is the `/results/data/`*`historyName`*`/`*`pointID`*`/`*`testName`*`/debug` directory of the ADE Explorer 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 any toolbar and choose *ADE Result Directory* to make it visible.




  Click *XTerm* to open the terminal window from where you can access the `netlist` or `psf` directories for the data point.

- The relevant job policy settings set up in the ADE Explorer environment are copied to the debug environment.

- You cannot use the debug environment to debug results from OCEAN-based tests.


© 2023 Cadence Design Systems, Inc.

- 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. 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 Explorer 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 `ExplorerRun.0`, ADE Explorer overwrites the active setup with the setup information in `ExplorerRun.0`. Then, the changes in the debug environment setup are merged with the active setup as described below:

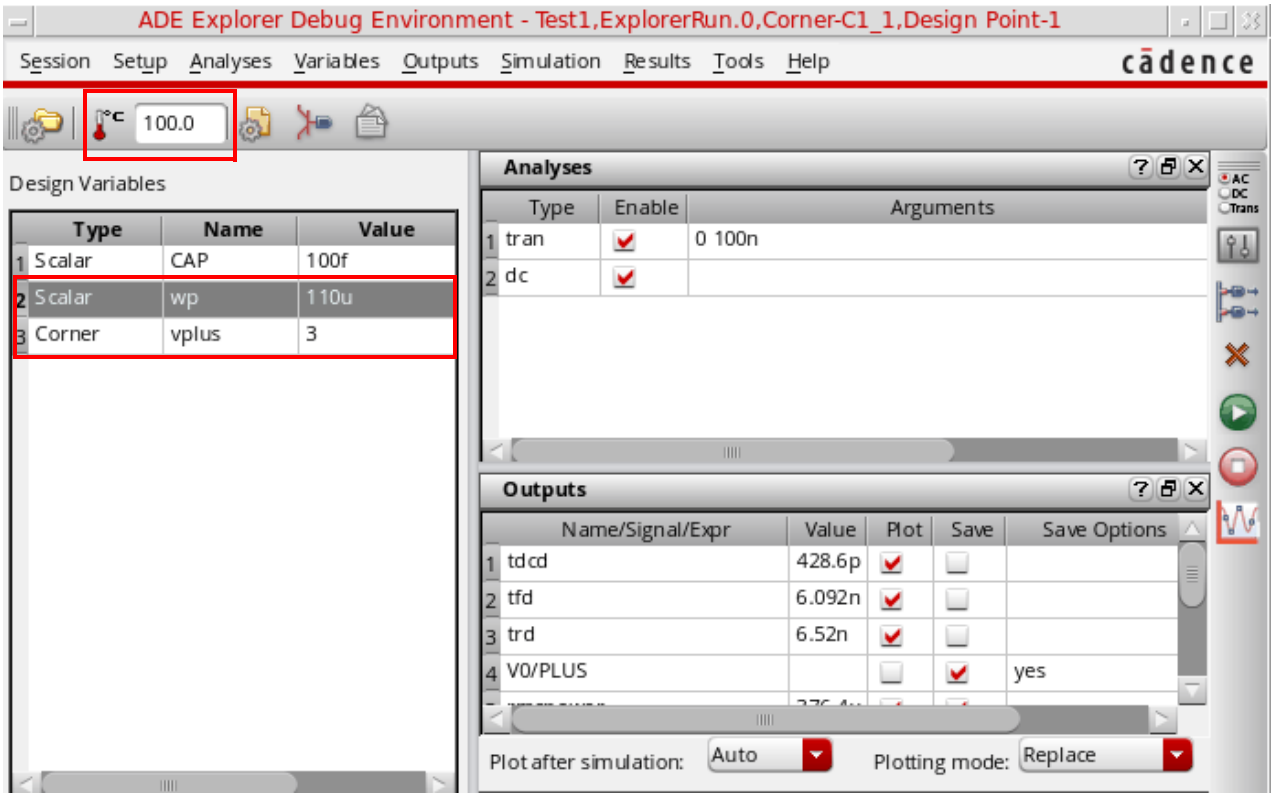
- Changes in design variables, analyses and outputs for a test in the debug environment are merged with the setup in the ADE Explorer view.
- Changes in scalar variables (global variables or device parameters with single value) in the debug environment overwrite the value of the corresponding global variables and device parameters in the ADE Explorer 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 Explorer view.
- Changes in corner values in the debug environment result in a new corner being created in the ADE Explorer view.

For example, if you choose *Session – Setup Back To ADE Explorer* after making the following changes in the debug environment, as shown in the figure:

- Modified the value of the scalar variable `wp` from `104u` to `110u`
- Modified the value of the corner variable `vplus` from `2.2` to `3`


© 2023 Cadence Design Systems, Inc.

- Modified the value of the *Temperature* variable for the corner from `27` to `100`.




The changes are merged into the active 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.


© 2023 Cadence Design Systems, Inc.

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.

© 2023 Cadence Design Systems, Inc.

© 2023 Cadence Design Systems, Inc.

# 8

# 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, your design must include devices with statistical models. Device parameters are modeled with statistical distributions which represent the variation of the fabrication process. Multiple simulations are performed during Monte Carlo analysis while randomly sampling from the statistical model distributions. Each sample point is simulated with different statistical parameter values based on the model distributions.

When the simulations finish, you can examine how the variation affects the overall yield of your design.

This chapter covers the following topics:


- Before Running a Monte Carlo Analysis on page 658
- Specifying Monte Carlo Run Options on page 660
- Specifying Instances for Monte Carlo Mismatch and Process Variation on page 692
- Running Advanced Methods on page 708
- Viewing Monte Carlo Results on page 714
- Managing Monte Carlo Results in the Yield View on page 715
- Generating Plots, Tables, Files, and Reports on page 720
- Viewing Statistical Parameters for Monte Carlo Samples on page 741



© 2023 Cadence Design Systems, Inc.

# Before Running a Monte Carlo Analysis

*Important*

The full-functionality of Monte Carlo analysis run is available only with the following simulators:

- Spectre circuit simulator
- Virtuoso Accelerated Parallel simulator (APS)
- Spectre AMS Designer and Xcelium Mixed-Signal simulator with Spectre or APS as the solver

  To select a solver, in the Setup pane, right-click 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 Spectre AMS Designer and Xcelium Mixed-Signal simulator.
  - You cannot use Spectre AMS Designer and Xcelium Mixed-Signal 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.

Your design must include devices or device models for which you have specified statistically varying parameter values. 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 Monte Carlo in the *Spectre Classic Simulator, Spectre APS, Spectre X, and Spectre XPS User Guide*.

Before you proceed with the Monte Carlo run, ensure the following:

1. The simulator specified is Spectre, APS, or AMS Designer (with Spectre as the solver). You can run Monte Carlo analysis only with these simulators.
2. 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. Therefore,


© 2023 Cadence Design Systems, Inc.

ensure that the *Plot* check box is selected for the output expressions that you want to be evaluated.

3. (Optional) If you have corners defined, you can also select the *Corners* check box on the Explorer Run Summary assistant or Setup assistant to run Monte Carlo over them.

   **Note:** You can change the corners you have defined on the Corners Setup form.

4. (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.

Monte Carlo sampling cannot be run in the following cases:

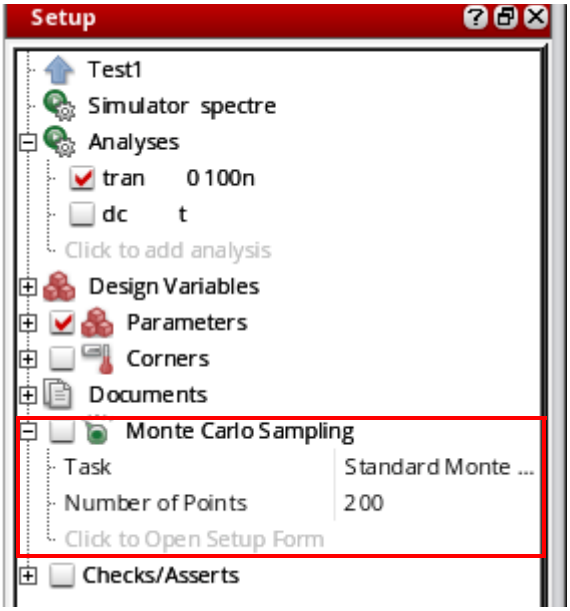
- Sweeps and corners are defined using the same variable
- Statistical corners are specified


© 2023 Cadence Design Systems, Inc.

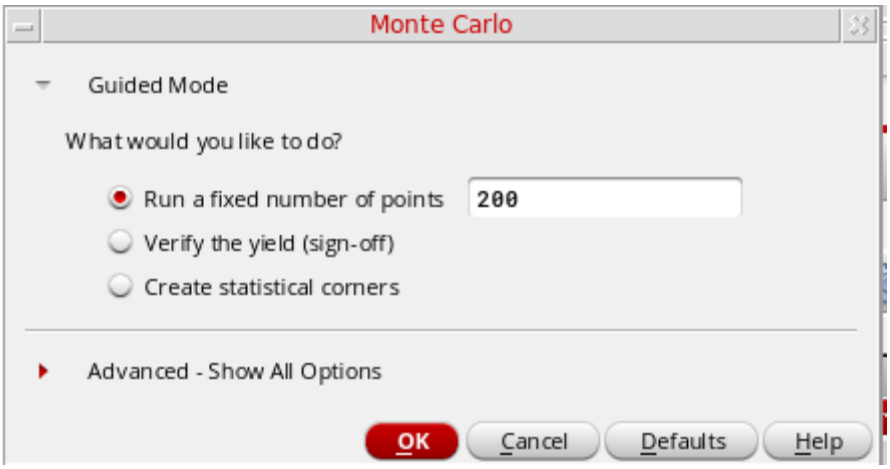
# Specifying Monte Carlo Run Options

To specify Monte Carlo run options, do one of the following:

- Choose *Tools – Monte Carlo*.
- In the Setup assistant, in the *Monte Carlo Sampling* section, click the `Click to Open Setup Form` text.




The Monte Carlo form appears as shown in the figure below. This form includes two sections, *Guided Mode* and *Advanced – Show All Options*.

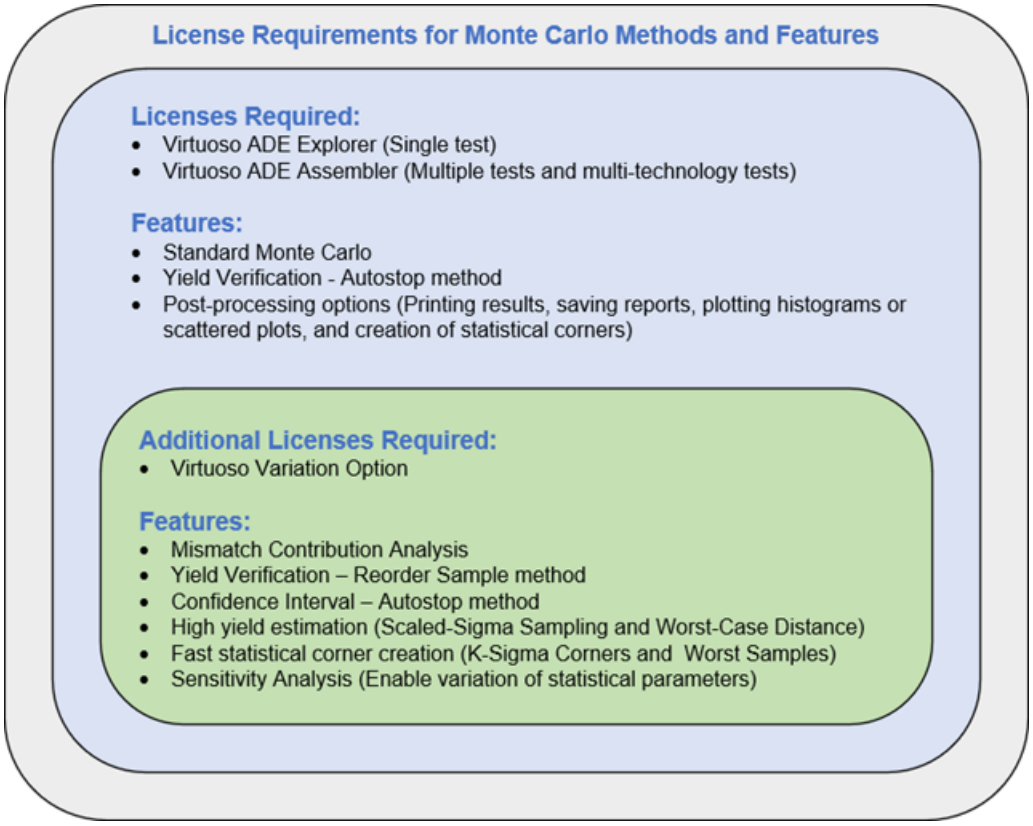




© 2023 Cadence Design Systems, Inc.

*Important*

With ADE Explorer license, the Standard Monte Carlo run is performed by default. If you have the Virtuoso Variation Option (VVO) license available, you can also run Monte Carlo using the advanced statistical analysis methods—`Sensitivity Accuracy`, `Mismatch Contribution`, `Yield Verification - Reorder Samples`, `Scaled-Sigma Sampling`, `Worst Case Distance`, `KSigma Corners`, `Worst Samples`, and `Confidence Interval - Autostop`. When running these advanced methods in ADE Explorer, the VVO license is checked out along with the ADE Explorer license. When you do not have the VVO license, you can run only Standard Monte Carlo and `Yield Verification - Autostop` methods with the ADE Explorer license.




**Note:** Before the VVO simulation starts, a lint check is performed that checks out the VVO license and then immediately checks it in. If the VVO license is not available, a warning message is displayed. This lint check is important because if it is not performed, ADE Explorer detects the license non-availability issue only after the VVO history has been created, which results in creating a blank history.

This section includes the following topics:


© 2023 Cadence Design Systems, Inc.

- Performing a Standard Monte Carlo Run on page 663
- Performing Sensitivity Accuracy on page 670
- Performing Yield Verification on page 673
- High Yield Methods on page 678
- Creating Statistical Corners on page 684
- Creating Statistical Corners on page 684
- Running Confidence Interval - Autostop Method on page 691

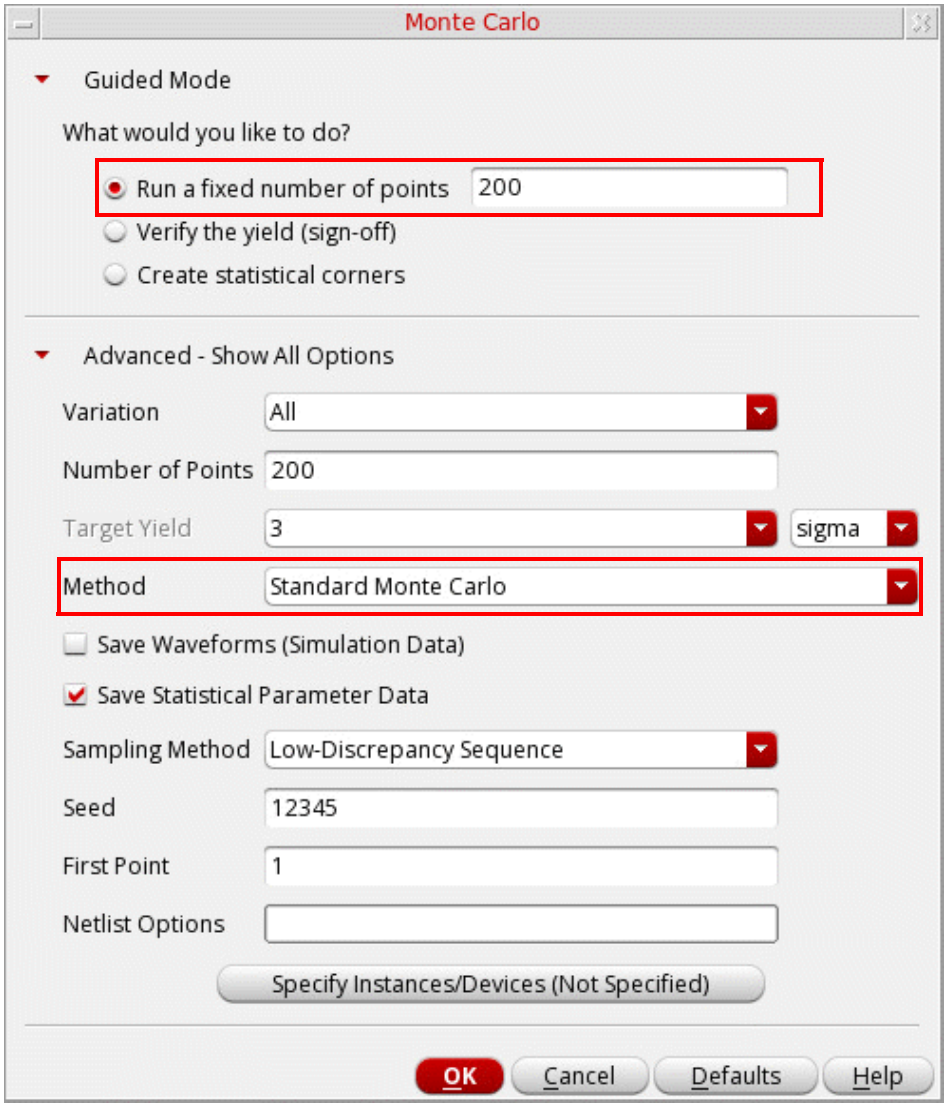

© 2023 Cadence Design Systems, Inc.

## Performing a Standard Monte Carlo Run

In Standard Monte Carlo, a specific number of points are simulated.

To perform a standard Monte Carlo run, perform the following steps in the Monte Carlo form.

1. In the *Guided Mode* section, select the *Run a fixed number of points* option. This option is selected by default on the Monte Carlo form.
2. In the text field, specify the number of points to be run. The default number of points is `200`.
3. When you expand the *Advanced - Show All Options* button, the additional form fields appear as shown in the figure below:





© 2023 Cadence Design Systems, Inc.

4. In the *Variation* field, select one of the statistical variation options:

| | |
|---|---|
| *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. In addition, 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 *Spectre Circuit Simulator User Guide*.

5. The *Number of Points* field shows the fixed number of points you specify in the *Guided Mode* section.

6. The *Target Yield* shows the target yield and sigma value. This field is disabled when you perform a standard Monte Carlo run.

7. In the *Method* drop-down, select the method you want to use to run Monte Carlo sampling. The default method is `Standard Monte Carlo Sampling`. The other available methods are:

   - `Sensitivity Accuracy`—For more information, see Performing Sensitivity Accuracy on page 670.
   - `Yield Verification - Autostop`—For more information, see Performing Yield Verification on page 673.
   - `Yield Verification - Reorder Samples`—For more information, see Performing Yield Verification on page 673.
   - `K-Sigma Corners`—For more information, see Creating Statistical Corners on page 684.
   - `Worst Samples`—For more information, see Creating Statistical Corners on page 684
   - `Confidence Interval`—For more information, see Running Confidence Interval - Autostop Method on page 691

8. Select the *Save Waveforms (Simulation Data)* option if you want to save output data (`psf`) for every Monte Carlo iteration so that you can perform post-processing operations (like plotting, printing, annotation, and re-evaluation) on individual iterations. Histograms


© 2023 Cadence Design Systems, Inc.

are always available for plotting even if this option is not selected. It is suggested not to select this option if you do not want to plot waveforms as it helps in saving disk space.

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.

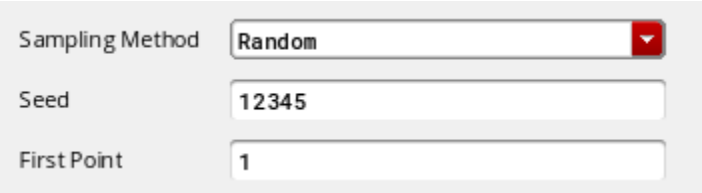
9. Select the *Save Statistical Parameter Data* option if you want to save statistical parameter information in the `maestro` view database. This option enables the post-processing of Monte Carlo results. For example, for mismatch contribution analysis, creation of statistical corners requires saved statistical parameter data.

10. In the *Sampling Method* drop-down, select one of the following statistical sampling methods 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`.


© 2023 Cadence Design Systems, Inc.

When the selected methods are `Random` and `Low-Discrepancy Sequence`, specify the following form fields:

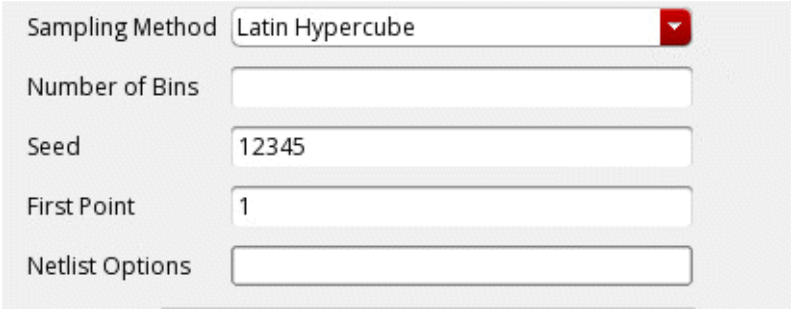



**a.** In the *Seed* field, specify a seed for the Monte Carlo analysis. By always specifying the same seed, you can reproduce a previous experiment. If you do not specify a seed, the value `12345` is used.

**b.** In the *First Point* field, specify a starting run number. The first point 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).

To reproduce a run or sequence of runs, you need to specify the same value in the *Starting Run Number* and the *Monte Carlo Seed* fields.

When the selected sampling method is `Latin Hypercube`, specify the number of bins (subdivisions) in the *Number of Bins* field.




**11.** In the *Netlist Options* field, specify additional analysis options that you want to generate in the netlist.

For example, "`nullmfactorcorrelation=yes`".

**12.** Click the *Specify Instances/Devices (Not Specified)* button to specify the sensitive instances and devices you want to either include or exclude for applying mismatch variations. For more information, see Specifying Instances for Monte Carlo Mismatch and Process Variation on page 692.

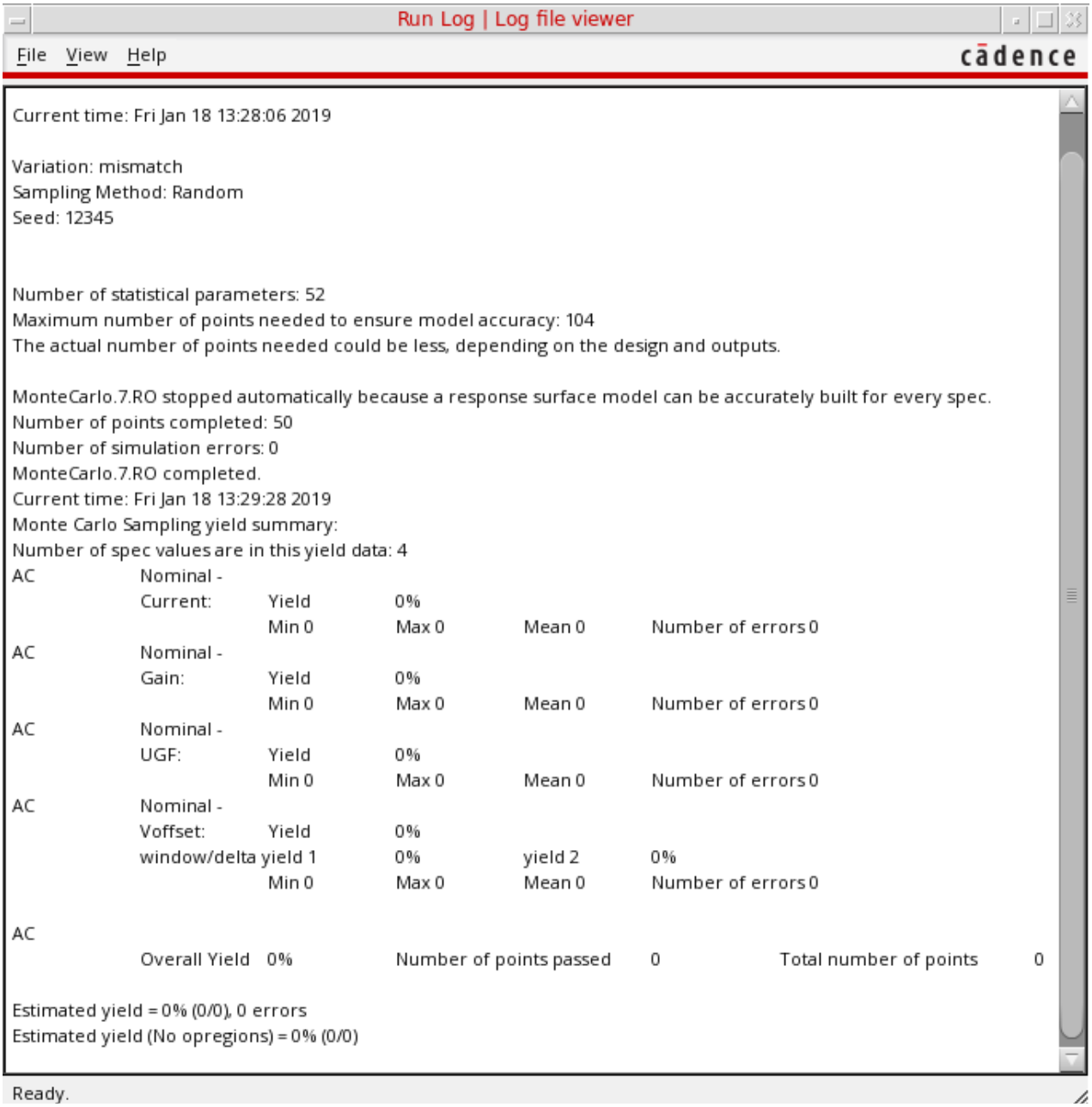
**13.** Click *OK*.

Now, click the run button to run the standard Monte Carlo sampling.


© 2023 Cadence Design Systems, Inc.

ADE Explorer 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.

When the run is in progress, Monte Carlo displays the details of the run in a log window, as shown below.




After the run is complete, ADE Explorer displays the results in the `Yield` view on the *Results* tab. Results are reported for each test, corner and measurement expression

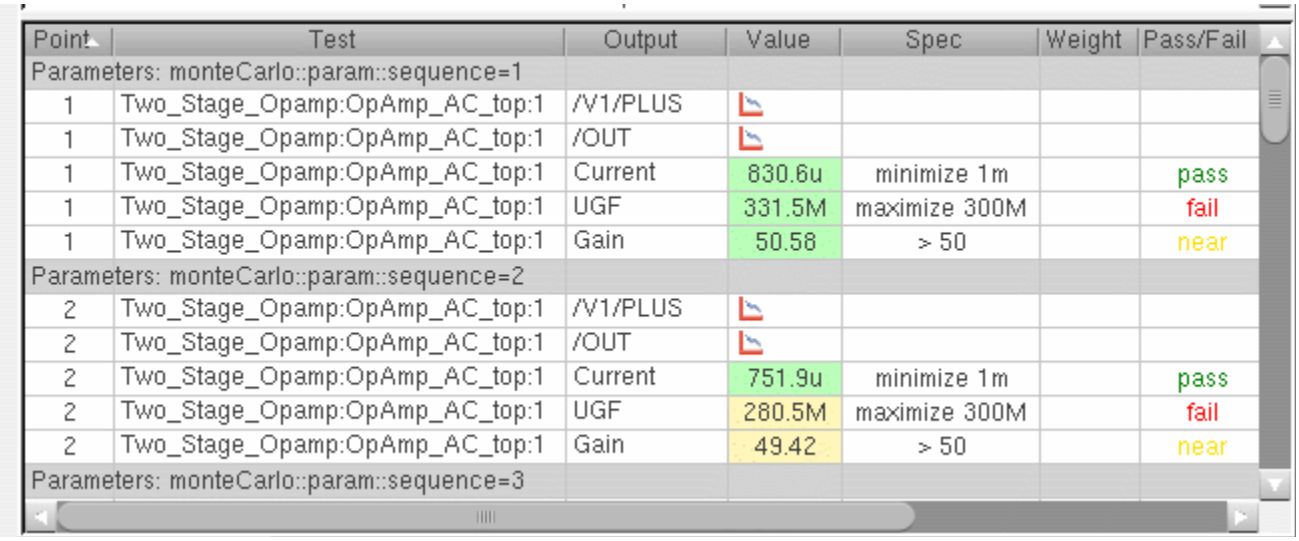

© 2023 Cadence Design Systems, Inc.

and the total yield is reported for the circuit. For more details, refer to Viewing Monte Carlo Results.

**Note:** You can change the default results view for the Monte Carlo Sampling run mode using the `defaultResultsViewForMonteCarlo` environment variable.

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.



| Point | Test | Output | Value | Spec | Weight | Pass/Fail |
|---|---|---|---|---|---|---|
| Parameters: monteCarlo::param::sequence=1 | | | | | | |
| 1 | Two_Stage_Opamp:OpAmp_AC_top:1 | /V1/PLUS | | | | |
| 1 | Two_Stage_Opamp:OpAmp_AC_top:1 | /OUT | | | | |
| 1 | Two_Stage_Opamp:OpAmp_AC_top:1 | Current | 830.6u | minimize 1m | | pass |
| 1 | Two_Stage_Opamp:OpAmp_AC_top:1 | UGF | 331.5M | maximize 300M | | fail |
| 1 | Two_Stage_Opamp:OpAmp_AC_top:1 | Gain | 50.58 | > 50 | | near |
| Parameters: monteCarlo::param::sequence=2 | | | | | | |
| 2 | Two_Stage_Opamp:OpAmp_AC_top:1 | /V1/PLUS | | | | |
| 2 | Two_Stage_Opamp:OpAmp_AC_top:1 | /OUT | | | | |
| 2 | Two_Stage_Opamp:OpAmp_AC_top:1 | Current | 751.9u | minimize 1m | | pass |
| 2 | Two_Stage_Opamp:OpAmp_AC_top:1 | UGF | 280.5M | maximize 300M | | fail |
| 2 | Two_Stage_Opamp:OpAmp_AC_top:1 | Gain | 49.42 | > 50 | | near |
| Parameters: monteCarlo::param::sequence=3 | | | | | | |

**Note:** The program converts waveforms that have `mcparamset` as the sweep name to histograms and plots each such waveform in a separate subwindow.

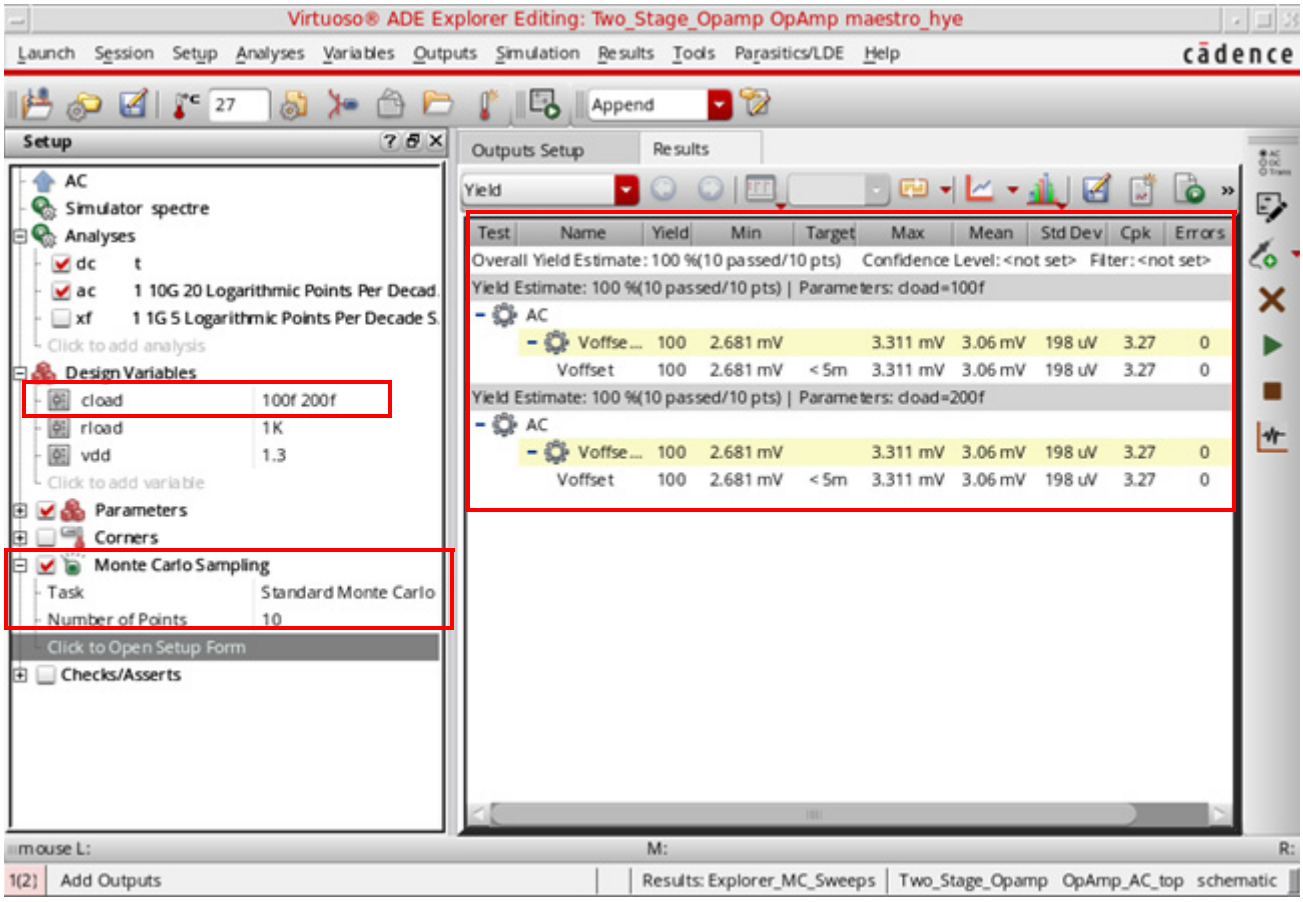
The program also displays a pass/fail type of histogram plot for the results. For more information about plotting histograms for Monte Carlo results, see Plotting Histograms on page 726.

### Running Monte Carlo with Sweep Values

If you have sweep values specified for design variables or parameters in the ADE Explorer setup, you can run Monte Carlo only using the Standard Monte Carlo method. If you run Monte Carlo using any other advanced method with sweep values enabled, a pop-up error message is displayed.


© 2023 Cadence Design Systems, Inc.

Consider the example below in which you have specified sweep values for design variable, `cload=100f 200f`. The method selected in the Monte Carlo options is Standard Monte Carlo. Now, when you run the simulation, the results for both sweep values appear on the *Results* tab, as shown in the figure below.




The yield results view contain separate rows for each swept variable or parameter. The overall yield is also displayed at the top.


© 2023 Cadence Design Systems, Inc.

## Performing Sensitivity Accuracy

**Note:** The Sensitivity Accuracy and mismatch contribution methods are available only when you have the Virtuoso Variation Option (VVO) license.

This method is used to post-process the Monte Carlo results and view the Mismatch Contribution table when you are not sure about the number of samples required to view the mismatch contribution results. This method helps in automatically determining the number of samples required to be 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.

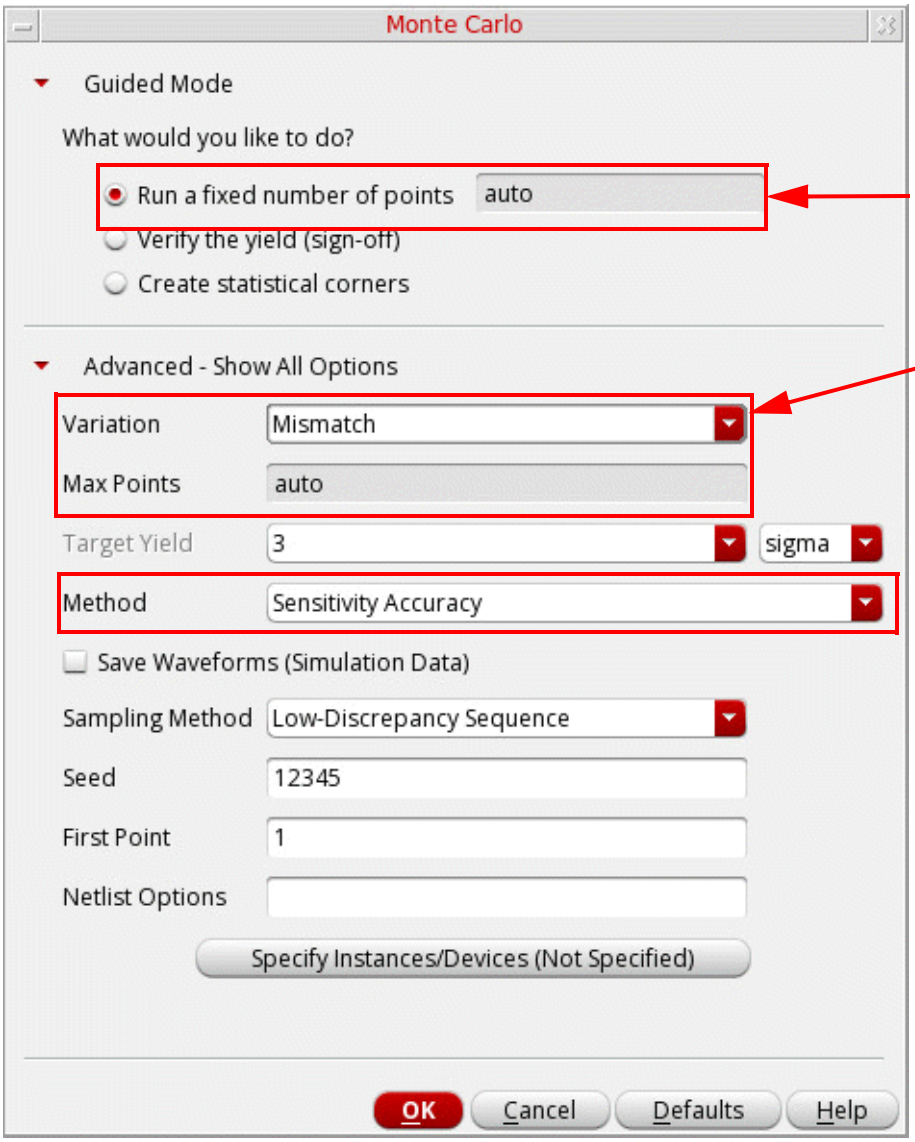
When you choose this method, the Monte Carlo run is stopped as soon as a model for variation contribution can be built for every output.

To run sensitivity accuracy method, ensure that *Run a fixed number of points* field is selected in the *Guided Mode* section of the Monte Carlo form.

➡ In the *Advanced - Show All Options* section, in the *Method* drop-down, select the `Sensitivity Accuracy` method.


© 2023 Cadence Design Systems, Inc.

The figure below shows the form fields that appear when the `Sensitivity Accuracy` method is selected.




Fixed number of points changed to

Select **Variation** as **Mismatch**

**Max Points** field added to show maximum number of points to be run.

The **Saved Statistical Parameter Data** option is no longer available as this data

Specify the following field values:

1. In the *Variation* drop-down, select variation as `Mismatch`.
2. In the *Max Points* field, specify the maximum number of points to be run. By default, this field is set to `1000`.
3. Select the *Save Waveform (Simulation Data)* if you want to save the waveform data (`psf` files) for plotting.


© 2023 Cadence Design Systems, Inc.

4. In the *Sampling Method* drop-down, select the method as `Random` or `Low-Discrepancy Sequence`. The `Latin Hypercube` sampling method is not available with Sensitivity Accuracy method.

   For more information about these form fields on this form, see Performing a Standard Monte Carlo Run on page 663.

### *Viewing Sensitivity Results*

You can view the sensitivity results to find out the sensitivity of specifications to variables, device parameters, statistical parameters and DC operating point parameters.

To view sensitivity results:

➡ Click the *Sensitivity Results* button on the toolbar in the *Results* tab.

In the Sensitivity Analysis window is displayed. In this window, the sensitivity data is displayed in a tabular format, where each data cell shows the sensitivity of a spec or parameter in the corresponding column to the spec or parameter displayed in the corresponding row. The specs or parameters displayed in the inputs and outputs depend on the selected tab.

For more information, see Sensitivity Analysis chapter in *Virtuoso ADE Assembler User Guide*.

### *Viewing Mismatch Contribution Results*

You can view the important contributors to mismatch variation by using the *Mismatch Contribution* post-processing command on the Monte Carlo results. It is recommended to choose the *Sensitivity Accuracy* method to automatically determine the number of required samples. After you run these required Monte Carlo samples, the *Mismatch Contribution* tab is visible. This tab is also visible when you manually specify the number of points instead of selecting the Sensitivity Accuracy method.

To view mismatch contribution results:

➡ Click *Mismatch Contribution* button on the toolbar in the *Results* tab.

The Sensitivity Analysis window is displayed showing the Mismatch Contribution tab.

For more information, see Mismatch Contribution Analysis in *Virtuoso Variation Option User Guide*.


© 2023 Cadence Design Systems, Inc.

## Performing Yield Verification

The accuracy of the yield estimate from a 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. You can perform one of the following yield verification methods to estimate the yield during the Monte Carlo analysis. These methods can be run when the target yield less than `4` sigma.

- `Yield Verification - Autostop`

  This method is used 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.

- `Yield Verification - Reorder Samples`

  *Important*

  This method is available only when you have the Virtuoso Variation Option (VVO) license.

  To verify the yield, you require a large number of samples when the yield is above or slightly below the target. For example, when verifying a 3-sigma target with confidence value 90 percent, the run can be stopped if the first 1800 samples (approximately) pass the specification. The sample reordering method helps you verify the yield without simulating such a large number of samples.

To perform yield verification using the `Yield Verification - Autostop` or `Yield Verification - Reorder Samples` methods,

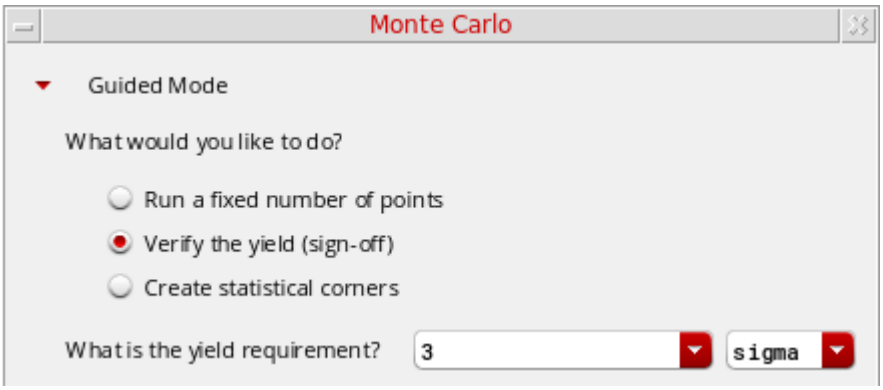
1. In the *Guided Mode*, select the *Verify the yield (sign-off)* option.

   When you select this option, the *What is the Yield requirement?* field appears in which you specify the sigma or percentage value. If you choose `sigma`, the default value is `3` and if you choose `%`, the default value is `99.865`. The value you specify in this field is always in synchronization with the *Target Yield* field available in the *Advanced - Show All Options* section.

   **Note:** These yield estimation methods are used only when the sigma value is less than `4`. For higher sigma values (`<4`), you can use scaled-sigma sampling or worst case


© 2023 Cadence Design Systems, Inc.

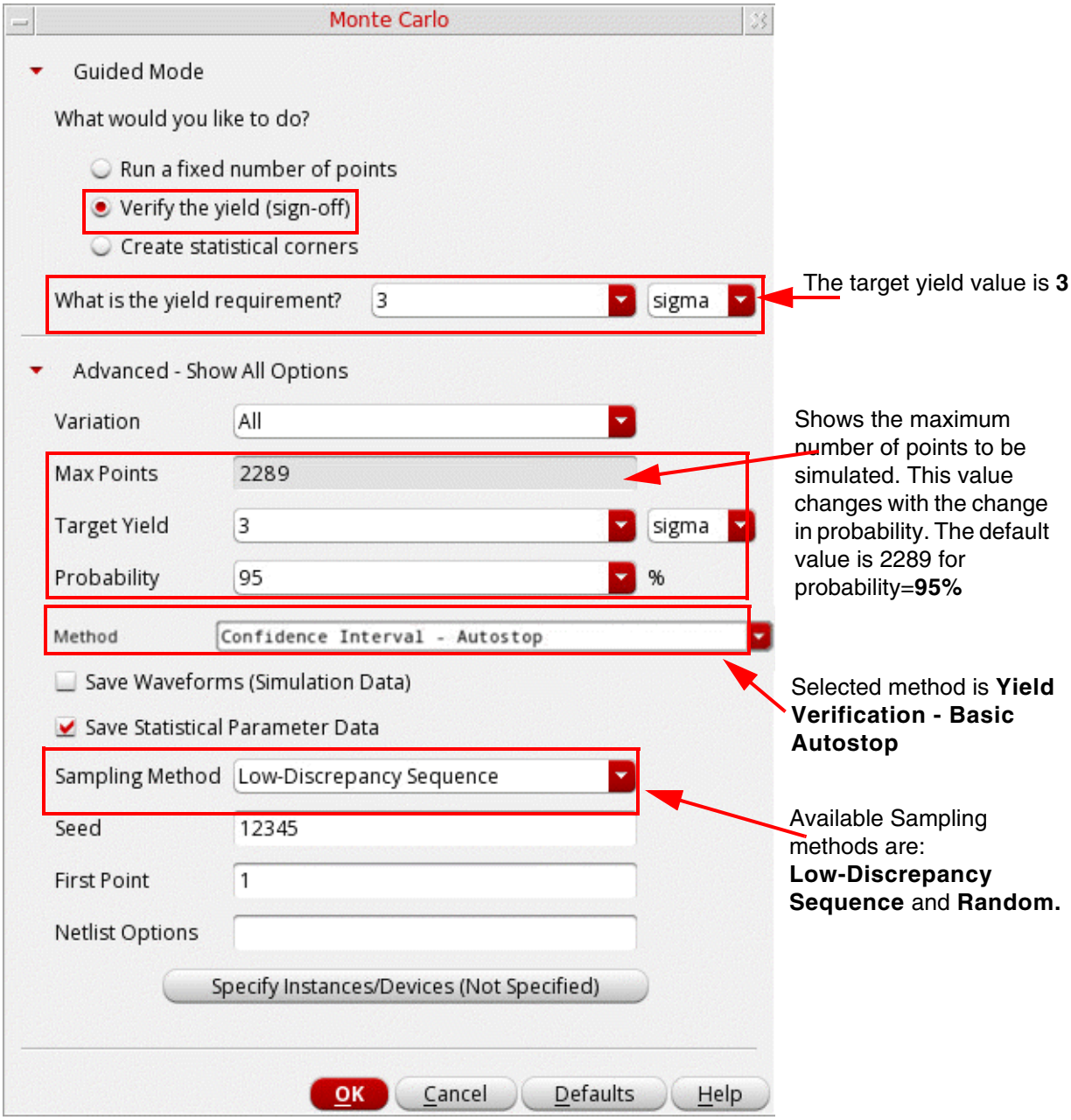
distance methods.





© 2023 Cadence Design Systems, Inc.

2. When the yield is `3`, in the *Advanced - Show All Options* section, specify the following fields:




   a. In the *Variation* drop-down, select variation as `All`, `Mismatch`, or `Process`.

   b. In the *Max Points* field, specify the maximum number of points to be run. By default, this field is set to `2289` for probability set to `95%`. The value of this field changes when you change the probability value.


© 2023 Cadence Design Systems, Inc.

**c.** In the *Target Yield* field, specify the yield value that you want to achieve for your design. The value of this field is same as the sigma field in the *Guided Mode* section.

**d.** In the *Probability* field, select a probability percentage value. 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 requires less simulations before auto stop is triggered. The default value is `95%`.

**3.** In the *Method* drop-down, select the method as `Yield Verification - Basic Autostop` or `Yield Verification - Sample Reorder`.

**Note:** If you have the Virtuoso Variation Option license available, `Yield Verification - Sample Reorder` method is selected by default when you select the *Verify the Yield Sign-off* option in *Guided Mode*.

**Note:** If you have the Virtuoso Variation Option License available, by default when you select the *Verify the Yield Sign-off* option in *Guided Mode*, `Yield Verification - Sample Reorder` method is selected. For more information about how to run this method, see Running the Yield Verification - Reorder Sample Method in *Virtuoso Variation Option User Guide*.

**4.** Select the *Save Waveforms (Simulation Data)* option if you want to save output data (`psf` files) for every Monte Carlo iteration so that you can perform post-processing operations (like plotting, printing, annotation, and re-evaluation) on individual iterations. Histograms are always available for plotting even if this option is not selected. It is suggested not to select this option if you do not want to plot waveforms as it helps in saving memory.

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 scalar results are calculated and saved for every run regardless of whether you select this check box or not.

**5.** Select the *Save Statistical Parameter* Data option if you want to save statistical parameter information in the ADE Explorer results database. This option enables the post-processing of Monte Carlo results. For example, mismatch contribution analysis, creation of statistical corners require saved statistical parameter data.

**Note:** The *Save Statistical Parameter Data* option is not available when the method is `Yield Verification - Sample Reorder`.


© 2023 Cadence Design Systems, Inc.

6. In the *Sampling Method* drop-down, select one of the following statistical sampling method to be used:

   - `Random`: This 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).
   - `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`.

     For more information about these methods, see Performing a Standard Monte Carlo Run on page 663.

7. In the *Seed* field, specify a seed for the Monte Carlo analysis. By always specifying the same seed, you can reproduce a previous experiment. If you do not specify a seed, the value `12345` is used.
8. In the *First Point* field, specify a starting run number. The first point 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).

   To reproduce a run or sequence of runs, you need to specify the same value in the *Starting Run Number* and the *Monte Carlo Seed* fields.
9. In the *Netlist Options* field, specify additional analysis options that you want to generate in the netlist.

   For example, "`nullmfactorcorrelation=yes`".
10. Click the *Specify Instances/Devices (Not Specified)* button to specify the sensitive instances and devices you want to either include or exclude for applying mismatch variations. For more information, see Specifying Instances for Monte Carlo Mismatch and Process Variation on page 692.
11. Click *OK*.


© 2023 Cadence Design Systems, Inc.

## High Yield Methods

*Important*

The high yield estimation methods are available only when you have the Virtuoso Variation Option (VVO) license.

While verifying the yield, when you specify a target yield greater than `4` sigma in the Monte Carlo form, the method in *Advanced - Show All Options* section is automatically changed to `Scaled-Sigma Sampling`. In addition, for high target yield values, the `Worst Case Distance` method is also available that you can select in the *Method* drop-down. For more information about these high yield estimation methods see the below sections:

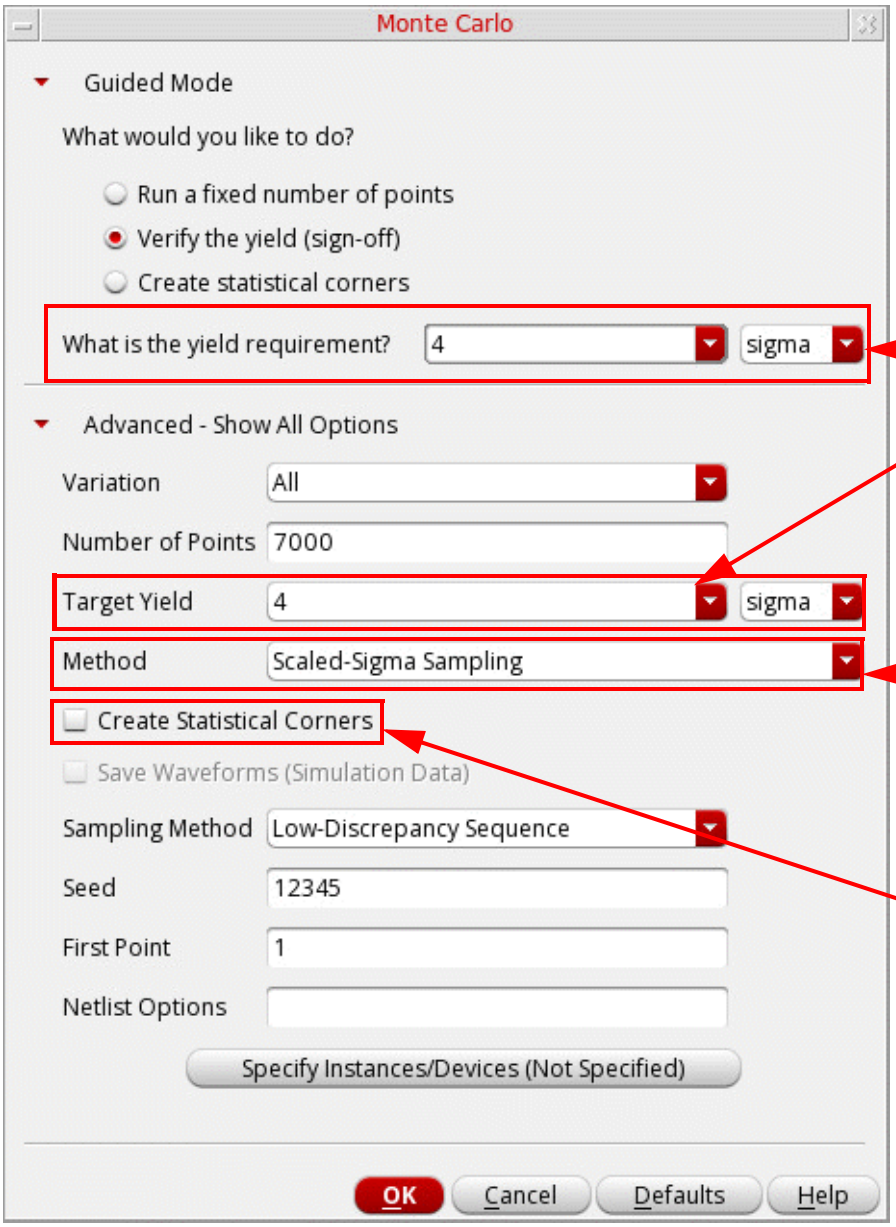
- Scaled-Sigma Sampling on page 678
- Worst Case Distance on page 680

### Scaled-Sigma Sampling

This method is selected by default when the yield is greater than `3` sigma. The Scaled-Sigma Sampling method generates samples where the standard deviation has been scaled up. As a result, a larger number of samples fall into the failure region of the distorted distribution. The failure rate is then estimated from the scaled samples.


© 2023 Cadence Design Systems, Inc.

The figure below shows the form fields that appear when you select this method:




The target yield (sigma value) is 4.

The **Number of Points** is set to **7000** by default.

The **Method** is changed to Scaled-Sigma Sampling

Select **Create Statistical Corners** check box if you want to create statistical corners. When this check box is selected, the task is changed to **Create Statistical corners** in the **Guided Mode**.

For detailed information about the common form-fields, see Performing a Standard Monte Carlo Run on page 663.

Note the following additional field values available for the Scaled-Sigma Sampling method:

- In the *Number of Points* field, specify the number of points you want to run. This field is set to `7000` by default.


© 2023 Cadence Design Systems, Inc.

- Select the *Create Statistical Corners* check box if you want to create statistical corners. When this option is selected, the task selection in the *Guided Mode* is changed to *Create statistical corners*.
- The *Sampling Method* drop-down lists the following three methods: `Low-Discrepancy Sequence` (default), `Latin Hypercube`, and `Random`.
- Click *OK*.

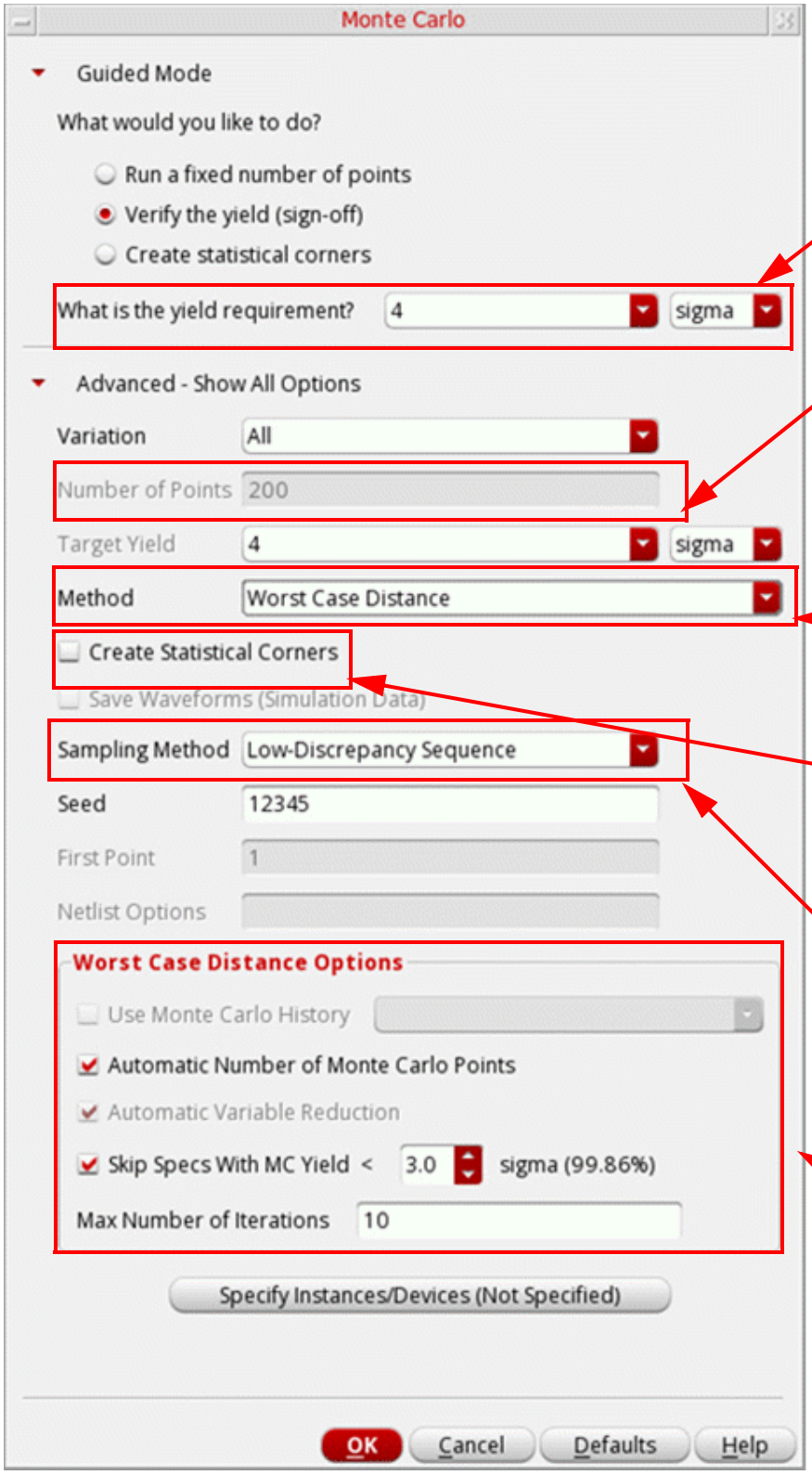
For more information, see The Scaled-Sigma Sampling Method in *Virtuoso Variation Option User Guide*.

### Worst Case Distance

When the target yield sigma value is greater than `4`, you can also select the Worst Case Distance method to verify the yield of the design. In this method, the high yield estimation finds the shortest distance, referred to as Worst Case Distance or WCD, from the nominal point to the specification boundary in the process/mismatch parameter space. For detailed information about this method, see

The figure below shows the form-fields that appear when the `Worst Case Distance` is selected in the *Method* drop-down.


© 2023 Cadence Design Systems, Inc.




The target yield (sigma value) is 4.

The default **Number of Points** is set to **200**. This field is available only when the **Automatic Number of Monte Carlo Points** check box in WCD Options is not selected.

The **Method** is changed to Scaled-Sigma Sampling

Select **Create Statistical Corners** check box if you want to create statistical corners. When this check box is selected, the task is changed to **Create Statistical corners** in the **Guided Mode**.

Three sampling methods available: **Low-Discrepancy Sequence**, **Random**, and **Latin Hypercube**.

Additional worst case distance options appear on the form.


© 2023 Cadence Design Systems, Inc.

For information about the common fields, see Performing a Standard Monte Carlo Run on page 663.

Note the following additional field values available for this method:

1. The *Number of Points* field is by default set to `200`. This field is disabled when the *Automatic Number of Monte Carlo Points* check box is selected.
2. Select the *Create Statistical Corners* check box if you want to create statistical corners. When this option is selected, the task selection in the *Guided Mode* is changed to *Create statistical corners*.
3. The *Sampling Method* drop-down lists the following three methods: `Low-Discrepancy Sequence` (default), `Latin Hypercube`, and `Random`.
4. In the *Worst Case Distance Options* section, specify the following fields:
    a. The *Use Monte Carlo History* option is enabled if you have already run a Monte Carlo simulation, you can use the process and mismatch data from the history of that run. In this drop-down list, you can select a reference Monte Carlo run history from the list of available histories. It is essential that the simulation data of the selected history contains the process and mismatch data. If any one of these data is not available, an error message is displayed.
    b. The *Automatic Number of Monte Carlo Points* check box is selected by default. This is used to enable the automatic selection of number of Monte Carlo points. When you select this check box, the *Number of Points* and *Automatic Variable Reduction* fields become unavailable. To manually provide the number of Monte Carlo points to be simulated, disable this check box.
    c. By default, the *Automatic Variable Reduction* check box is disabled. To enable this check box, deselect the *Automatically Select Number of Monte Carlo Points* check box. The *Automatic Variable Reduction* option reduces the set of statistical variables by eliminating insignificant variables (variables that have no variation or have no influence on the WCD point). Insignificant variables bring noise and require more simulations for sensitivity analysis. Therefore, it is recommended to enable variable reduction.
    d. To ignore the specifications for which Monte Carlo yield is less than a specified percentage, select the *Skip Specs With MC Yield <* check box and specify a sigma value. The default value of this field is `3` sigma; therefore, specifications for which the Monte Carlo yield is less than 99.86% are ignored. If you want to run high yield estimation on all the specifications, disable the *Skip Specs With MC Yield <* check box.


© 2023 Cadence Design Systems, Inc.

**e.** In the *Max Number of Iterations* field, specify the maximum number of iterations to be run for each specification. The default number of iterations is `10`.

**5.** Click *OK*.

For information, see Running the Worst-Case Distance Method in *Virtuoso Variation Option User Guide*.


© 2023 Cadence Design Systems, Inc.

## Creating Statistical Corners

**Note:** The methods to create statistical corners are available only when you have the Virtuoso Variation Option (VVO) license.

For target yield values greater than `3`, you can create statistical corners using the high yield estimation methods, `Scaled-Sigma Sampling` and `Worst Case Distance`. When the target yield is `3`, the advanced methods can be used to create statistical corners.

This section includes the following topics:

- Using K-Sigma Corners Method
- Using Worst Samples Method
- Creating Statistical Corners using High Yield Methods
- Creating Statistical Corners from Existing Run History

To create statistical corners,

➡ In the *Guided Mode* section, select the *Create Statistical Corners* check box and specify a target yield value.

When you select target yield less than `4`, you need to specify your requirements—whether you want speed or accuracy. Select one of the below listed options:

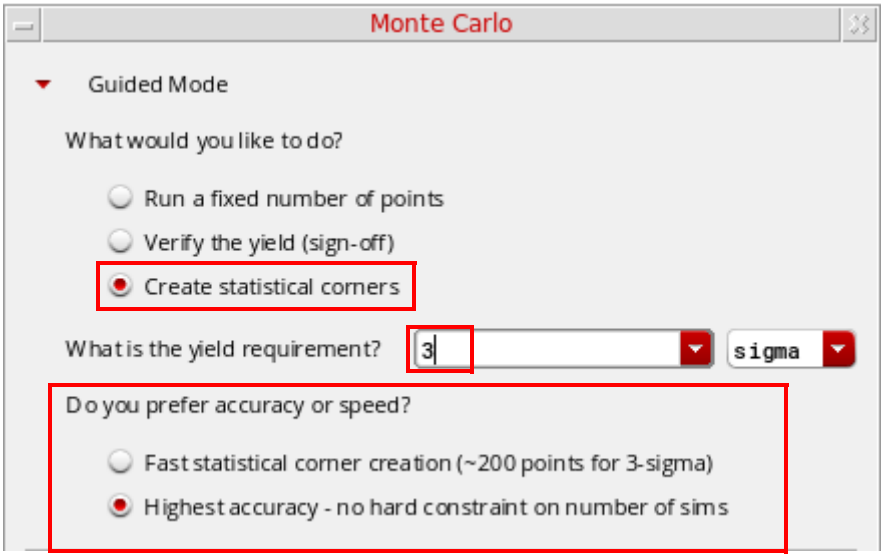
- *Fast statistical Corner creation (~200 points for 3-sigma)*—Select this option if you prefer speed over accuracy. When you select this option, the K-Sigma Corners method is performed to create statistical corners and the `K-Sigma Corners` method is selected in the *Method* drop-down. The K-Sigma Corners is strongly recommended over the Worst Samples method when the number of statistical parameters is large (> 1000).

  For information about how to use this method, see Using K-Sigma Corners Method on page 685.

- *Highest accuracy-no hard constraint on number of sims*—Select this option if you prefer accuracy over speed. When you select this option, the Worst Samples method is performed to create statistical corners and the `Worst Samples` method is selected in the *Method* drop-down.


© 2023 Cadence Design Systems, Inc.

For information about how to use this method, see Using Worst Samples Method on page 687.

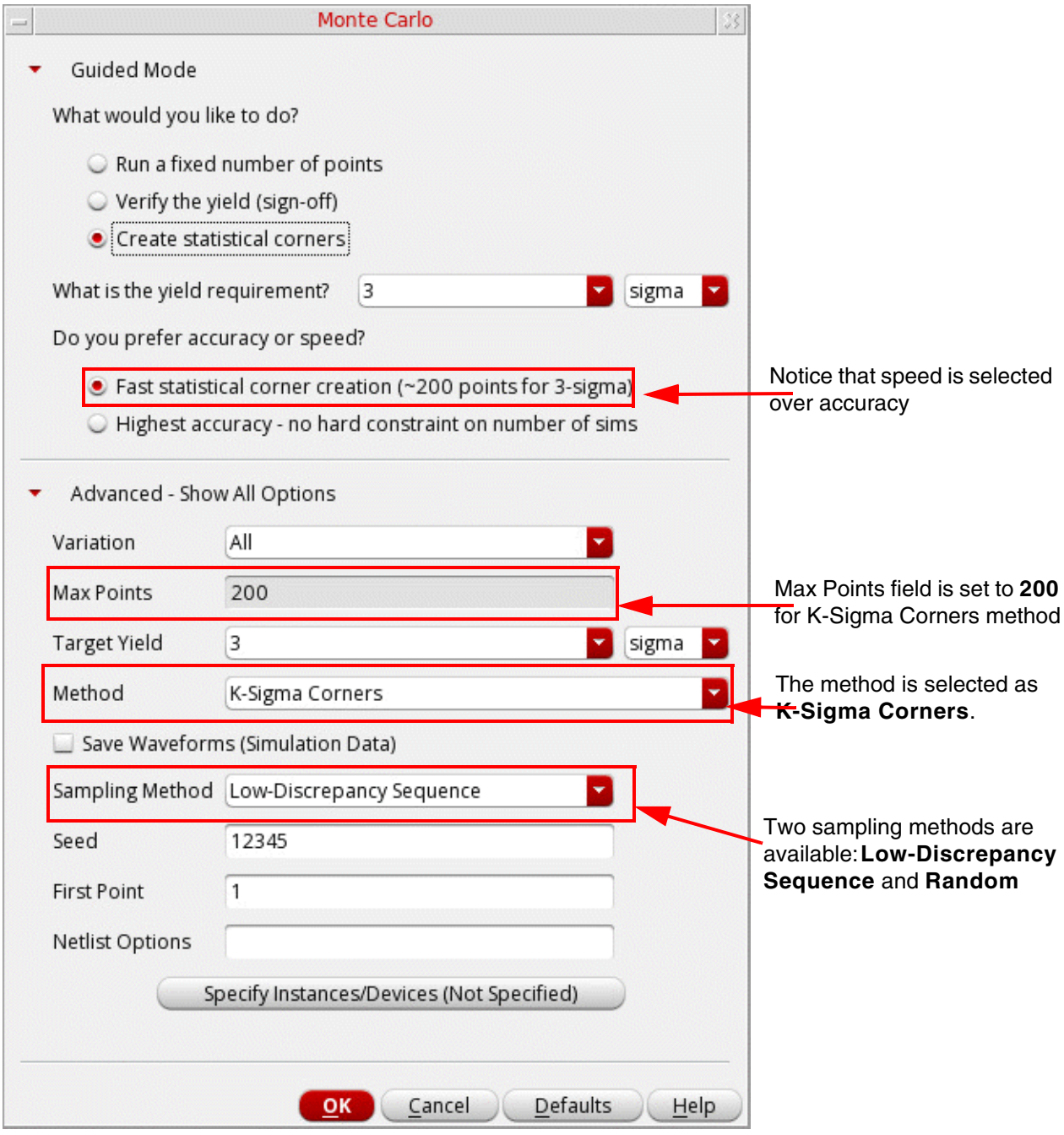



### Using K-Sigma Corners Method

This method creates the K-sigma statistical corner that meets the specified target yield value and applies the stopping criteria according to which the Monte Carlo simulation is stopped when the K-sigma statistical corner has been created for each specification. It is possible that multiple corners meet the target specification criteria. Therefore, the K-sigma method finds the most representative corner by calculating the minimum distance to the nominal point. This representative corner has a greater probability to occur. The statistical corner can then be used for further analysis of the design. For more information, see Statistical Corner Creation in *Virtuoso Variation Option User Guide*.


© 2023 Cadence Design Systems, Inc.

When you select the *Fast statistical Corner creation (~200 points for 3-sigma)* option, the K-Sigma corner method is performed. The figure below shows the form fields for this method:




For information about the common fields, see Performing a Standard Monte Carlo Run on page 663.


© 2023 Cadence Design Systems, Inc.

Note the following additional field values available for the `K-Sigma Corners` method:

- The maximum number of points (samples) to be simulated are displayed in the *Max Points* field. This value is automatically set to `200` for this method.
- The *Sampling Method* drop-down lists the following three methods: `Low-Discrepancy Sequence` (default), `Latin Hypercube`, and `Random`.
- Click *OK.*

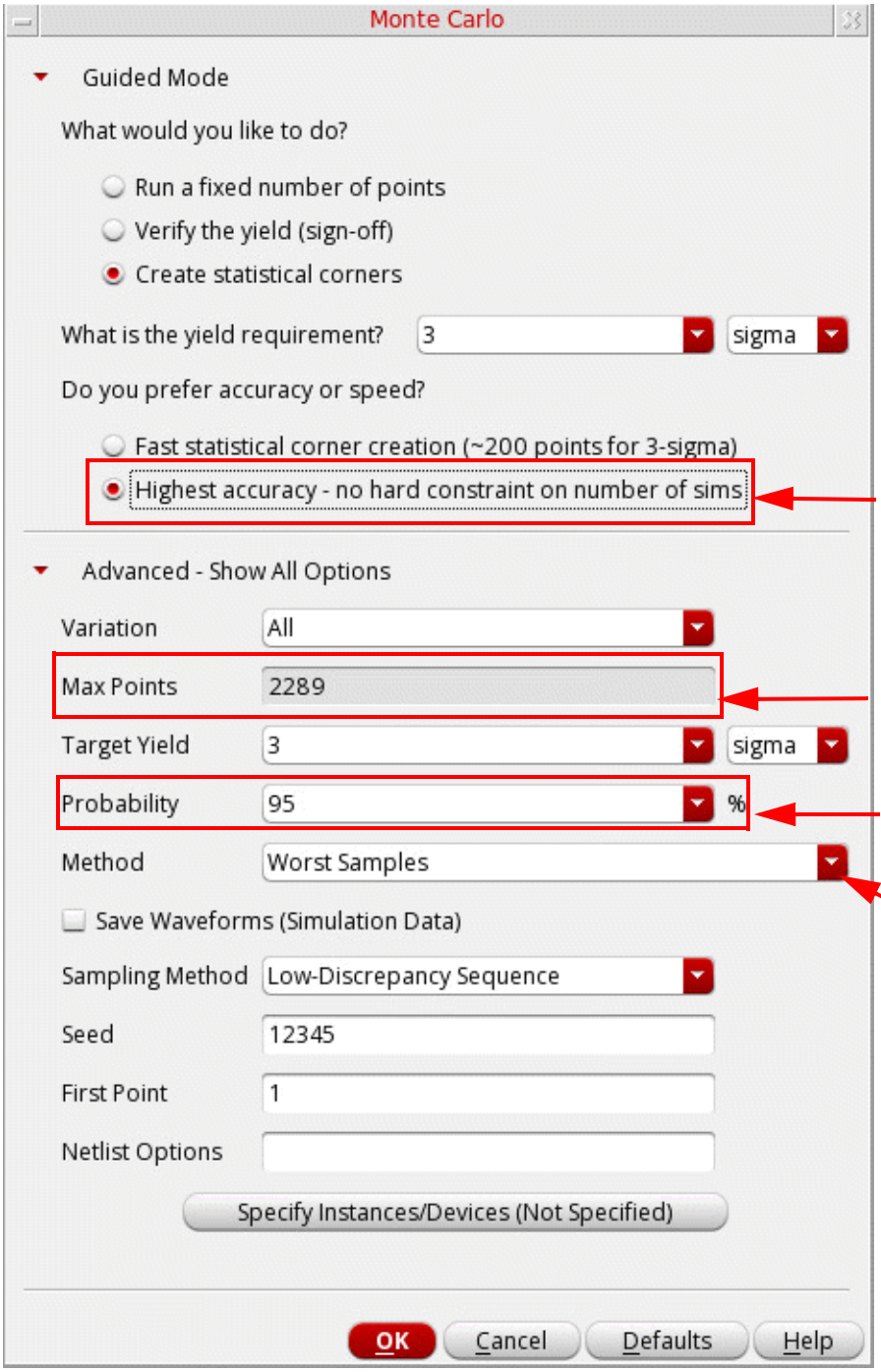
### Using Worst Samples Method

The `Worst Samples` method is recommended when high accuracy is needed and the number of statistical parameters is not large (< 1000).

In this method, the points are reordered based on the response surface modeling and sampling is stopped when the worst sample is found for each specification. The statistical corners are automatically created from the worst samples. For more information, see Statistical Corner Creation in *Virtuoso Variation Option User Guide*.


© 2023 Cadence Design Systems, Inc.

When you select the *Highest accuracy-no hard constraint on number of sims* option, the `Worst Samples` method is used to create statistical corners. The figure below shows the form fields for this method:




Notice that accuracy is selected over speed.

The maximum number of points is automatically set to **2289** for this method. This value changes when you change the probability.

Specify the probability (confidence level). The default probability is **95%**.

The method selected is **Worst Samples**.


© 2023 Cadence Design Systems, Inc.

For information about the common fields, see Performing a Standard Monte Carlo Run on page 663.

Note the following additional field values available for the `Worst Samples` method:

- The maximum number of points (samples) to be simulated are displayed in the *Max Points* field and automatically set to `2289` when the probability is `95%`. This value changes when you change the probability.
- In the *Probability* field, specify the probability value (confidence level). The default probability is `95%`.
- The *Sampling Method* drop-down lists the following three methods: `Low-Discrepancy Sequence` (default), `Latin Hypercube`, and `Random`.

For information about how to create statistical corner using the advanced methods, see Statistical Corner Creation in *Virtuoso Variation Option User Guide*.

### Creating Statistical Corners using High Yield Methods

**Note:** The high yield estimation methods are available only when you have the Virtuoso Variation Option (VVO) license.

While verifying the yield, when you specify a target yield greater than `3` sigma in the Monte Carlo form, the method in *Advanced - Show All Options* section is automatically changed to `Scaled-Sigma Sampling`. In addition, for high target yield values, the `Worst Case Distance` method is also available that you can select in the *Method* drop-down. For more information about these high yield estimation methods see the below sections:

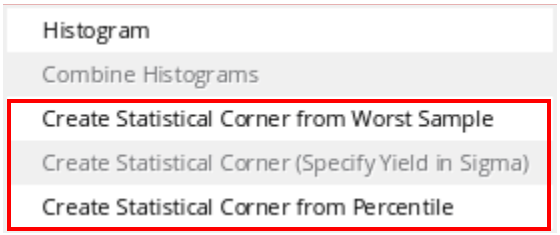
- Scaled-Sigma Sampling on page 678
- Worst Case Distance on page 680

### Creating Statistical Corners from Existing Run History

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.


© 2023 Cadence Design Systems, Inc.

To create statistical corners from the Monte Carlo results, right-click the specification row in the *Results* tab and choose one of the methods shown in the figure below.




- Creating Statistical Corner from Worst Sample

  This method identifies a sample that generated the worst result for a specification and creates a statistical corner using the sample details.

- Creating Statistical Corner (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.

- Creating Statistical Corner from Percentile

  This method is used to create a corner for the nth worst percentile for a given specification.

For more information about these methods, see Creating Statistical Corners in *Virtuoso ADE Assembler User Guide*.


© 2023 Cadence Design Systems, Inc.

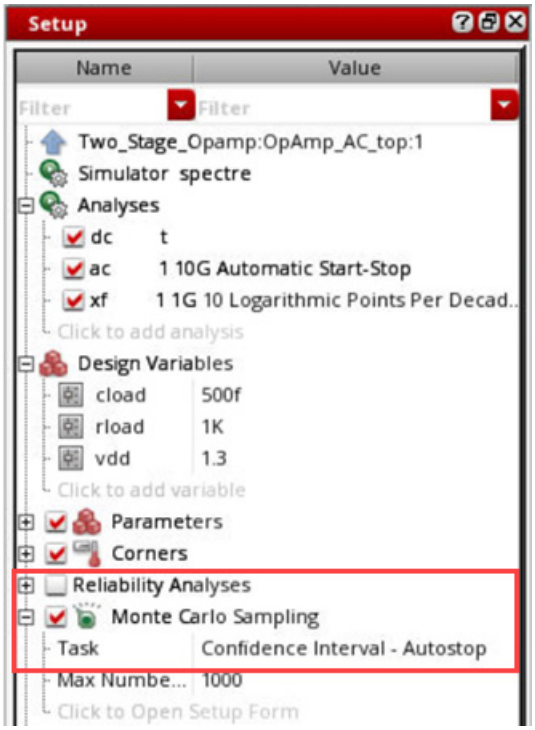
## Running Confidence Interval - Autostop Method

The *Confidence Interval - Autostop* method automatically stops Monte Carlo run when the confidence interval for the output standard deviation divided by the output value range falls below the *Stop Percentage* value. The default *Stop Percentage* value is `5%`.

In this method, you need not define number of sample points to run, target yield value, and specification targets because Monte Carlo run stops automatically when the stopping criteria is met, where stopping criteria is defined as follows:

$$\frac{(sigma_{upper\ bound} - sigma_{lower\ bound})}{(output_{upper\ bound} - output_{lower\ bound})} < Stop\ Percentage$$

To select this method, open the Monte Carlo form, and then choose `Confidence Interval - Autostop` from the *Method* drop-down list in the *Advanced - Show All Options* section.




For more information about this method, see The Confidence Interval - Autostop Method in *Virtuoso Variation Option User Guide*.


© 2023 Cadence Design Systems, Inc.

# Specifying Instances for Monte Carlo Mismatch and Process Variation

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 to which the variation must be applied. The variation will also be applied to all subcircuits instantiated under the selected instances.
- Select specific instances to the which variation must not be applied. The variation will not be applied to any subcircuit instantiated under the selected instances.

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.


© 2023 Cadence Design Systems, Inc.

The Specify Instances for Monte Carlo form appears.

2. In the *Variation Applied to Selected/Unselected Instances* group box, specify the following drop-down options:

| Drop-Down | Options |
|---|---|
| *Variation* | Specifies the type of variation to be applied.<br>■ *All*: To apply mismatch and process variations<br>■ *Mismatch*: To apply mismatch variations only<br>■ *Process*: To apply process variations only |


© 2023 Cadence Design Systems, Inc.

| Drop-Down | Options |
|---|---|
| *Selected* | Specifies the type of variation to be applied to the selected instances and devices.<br>■ *Mismatch and Process*: To apply mismatch and process variations to the selected instances and devices<br>■ *Mismatch*: To apply mismatch variations to the selected instances and devices<br>■ Process: To apply process variations to the selected instances and devices<br>■ *No Variation*: No variation to be applied to the selected instances and devices<br>**Note:** The options change based on the selection done in the *Variation* drop-down. |
| *Unselected* | Specifies the type of variation to be applied to the instances and devices that are not selected.<br>■ *Mismatch and Process*: To apply mismatch and process variations to the instances and devices that are not selected<br>■ *Mismatch*: To apply mismatch variations to the instances and devices that are not selected<br>■ Process: To apply process variations to the instances and devices that are not selected<br>■ *No Variation*: No variation to be applied to the instances and devices that are not selected |

**Note:** The options available in the drop-down lists depend on the capabilities provided by the Spectre simulator. Spectre 16.1 ISR5 or later provides an additional set of options compared to older Spectre versions.


© 2023 Cadence Design Systems, Inc.

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.<br><br>For more information, see Selecting Schematic Instances for Applying Mismatch Variations on page 695. |
| *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.<br><br>For more information, see Selecting Instances of Cellviews for Applying Mismatch Variations on page 698. |
| *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.<br><br>For more information, see Selecting Subcircuit Instances for Applying Mismatch Variations on page 701. |

*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:


© 2023 Cadence Design Systems, Inc.

1. Select the *Schematic* option.

   The test name is displayed in the *Test* drop-down list.

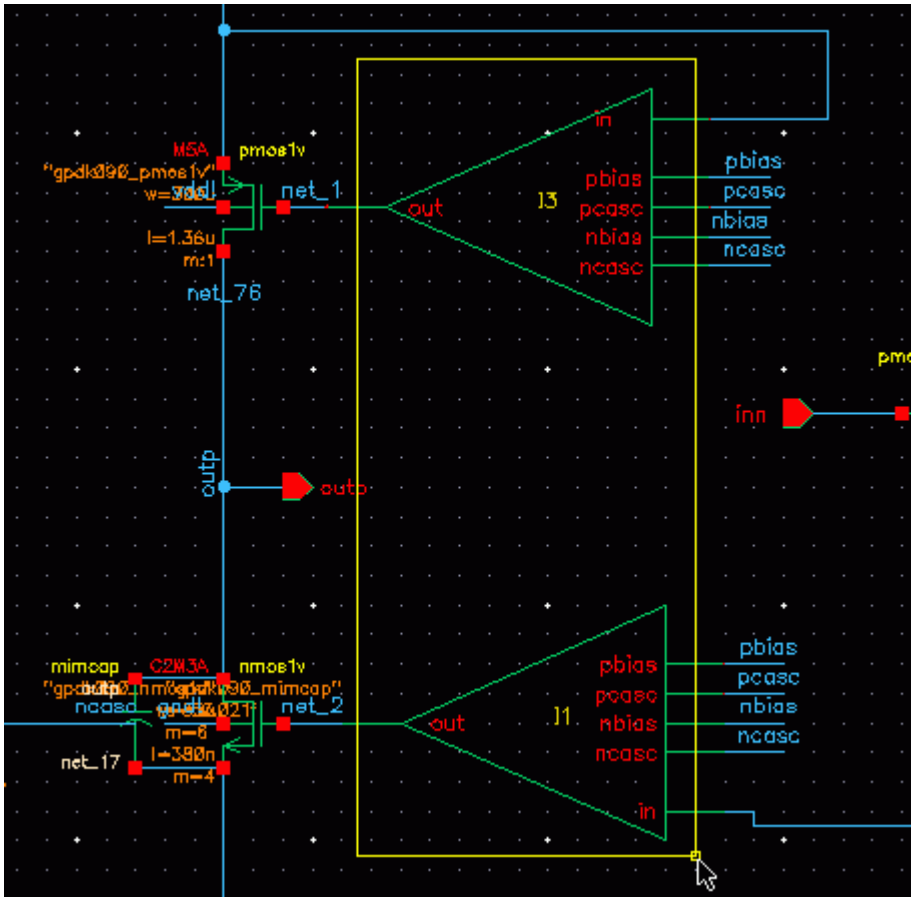
2. Click the *Select Instances* button. The schematic for the test is opened in a new tab.
3. In the schematic, select one or more instances. To select more than one instance at a time, do one of the following:

   - Hold down the *Shift* key and click 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.


© 2023 Cadence Design Systems, Inc.

In the following example, instances `I1` and `I3` that are within the yellow bounding box are included in the selection.




4. Press the *Esc* key when you are done.

   The selected instances are displayed in the *Choices* field and are simultaneously listed in the summary table at the bottom of the form.

5. In the *Choices* field, select the instances you want to include or exclude for applying mismatch variations.

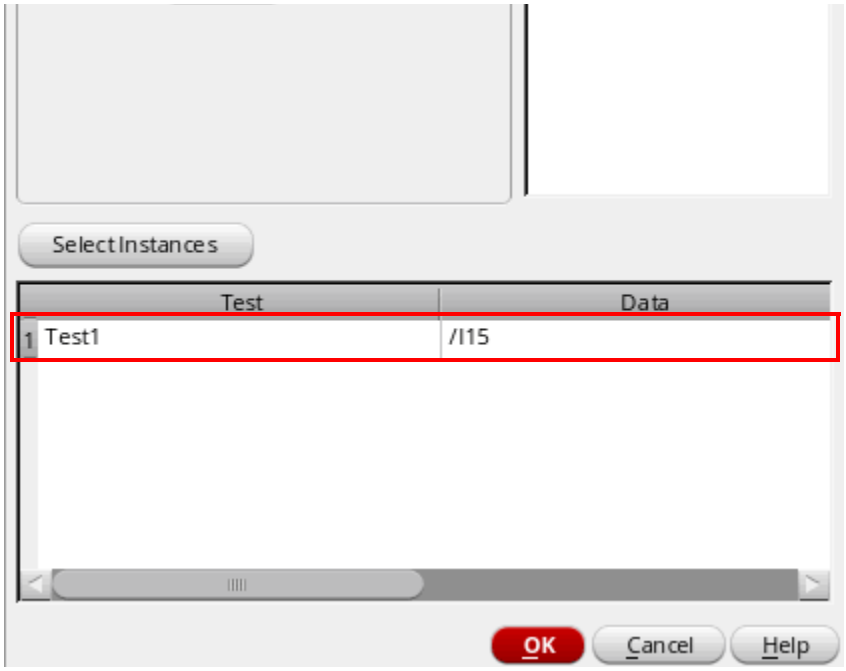
   To select multiple instances, hold down the *Shift* key (for contiguous selection) or the *Ctrl* key (for noncontiguous selection) and click the next instance to add it to the selection set.

   The instances listed in the *Choices* field are cross-linked with the schematic. When you click an instance name in *Choices*, it will be highlighted in the schematic and vice-versa.

6. Click *Apply*.


© 2023 Cadence Design Systems, Inc.

The selected instances are added for the test. For example, in the following figure shows the instance `/I9` added for the given test.

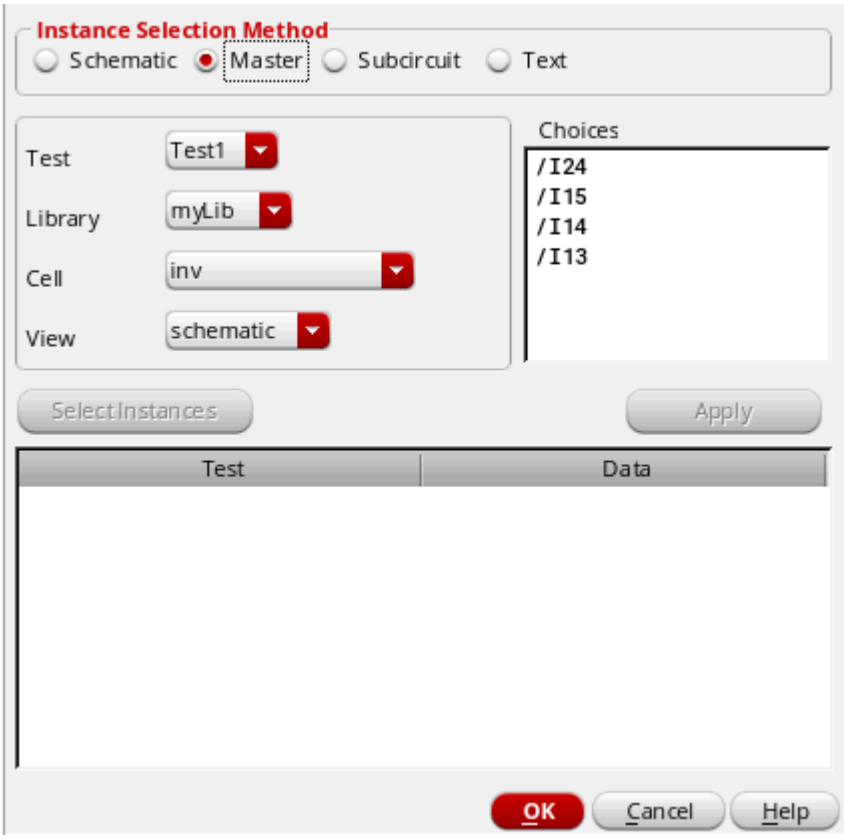



## Selecting Instances of Cellviews for Applying Mismatch Variations

To select instances of cellviews you want to include or exclude for applying mismatch variations, do the following:


© 2023 Cadence Design Systems, Inc.

1. Select the *Master* option.




The *Test* drop-down list displays the test name opened in the ADE Explorer setup.

2. Use the *Library*, *Cell* and *View* drop-down lists to select the library, cell and view in which the cellview exists.

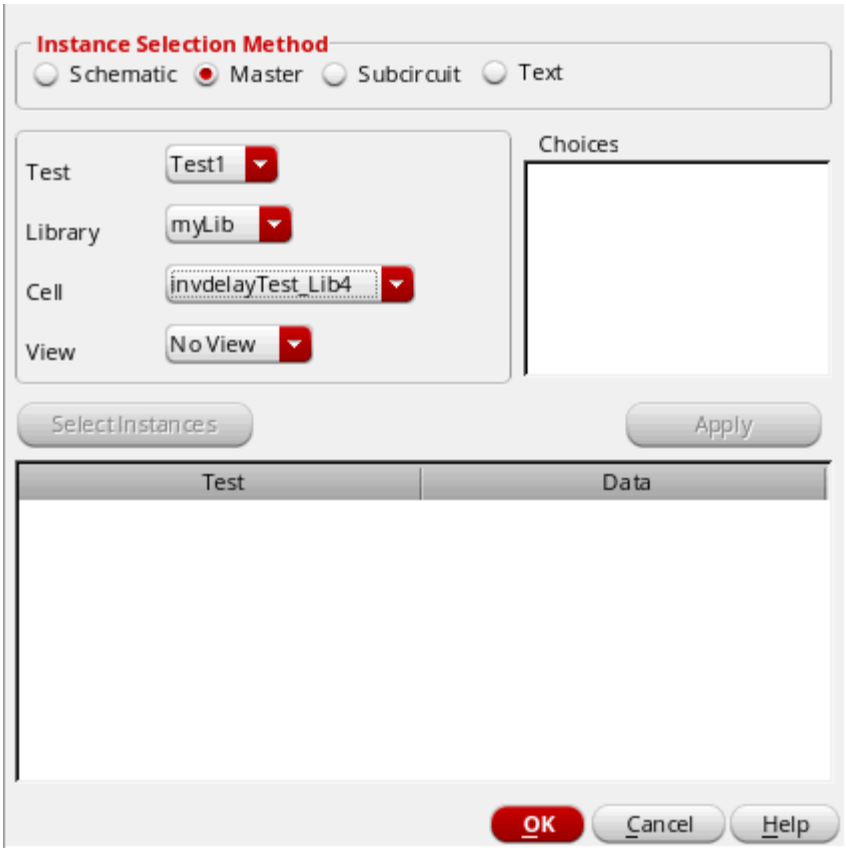
   All the instances of the cellview in the design for the test are displayed in the *Choices* field.

   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.
   - You can add only instances of cellviews that exist in the design. If instances of a cellview do not exist in the design, the *View* field displays the text *No View* indicating that you cannot select instances of the cellview.


© 2023 Cadence Design Systems, Inc.

For example, in the following figure, the *View* field displays the text *No View* because none of the cellviews of the `invdelayTest_Lib4` cell in the `myLib` library are instantiated in the design for the given test.



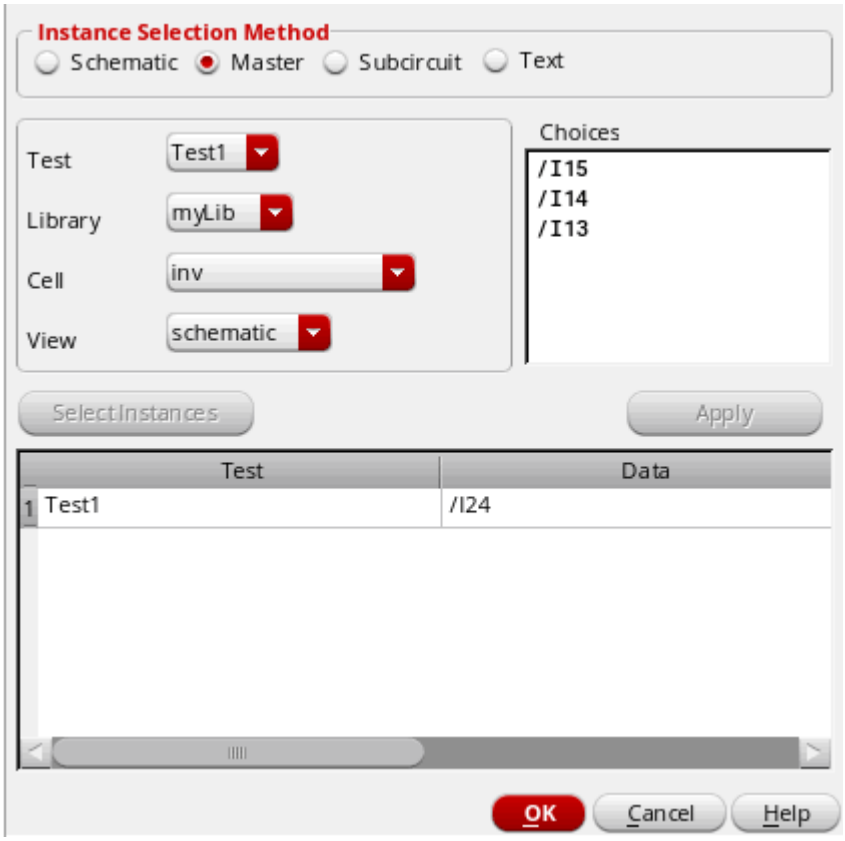

3. In the *Choices* field, select the cellview instances you want to include or exclude for applying mismatch variations.

   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.

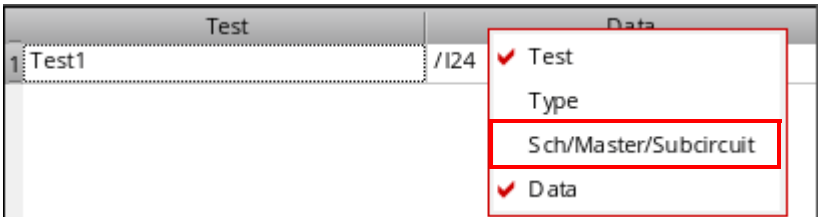
4. Click *Apply*.


© 2023 Cadence Design Systems, Inc.

The selected cellview instances are added for the test. For example, the following figure shows that the cellview instance `/I24` has been added for the given test.




**Note:** By default, the *Type* and *Sch/Master/Subcircuit* columns are not visible. To view these columns, right-click the title bar and select the name of the column to be displayed in this table.

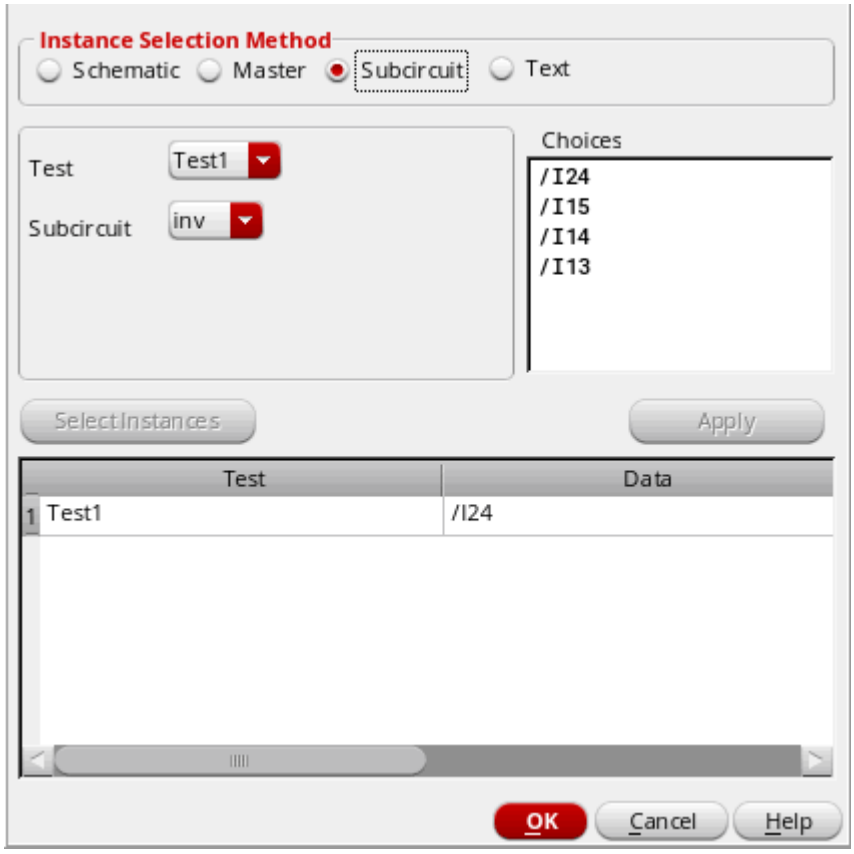



## Selecting Subcircuit Instances for Applying Mismatch Variations

To select the subcircuit instances you want to include or exclude for applying mismatch variations, do the following:


© 2023 Cadence Design Systems, Inc.

1. Select the *Subcircuit* option.




   The test name is displayed in the *Test* drop-down list.

2. 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.

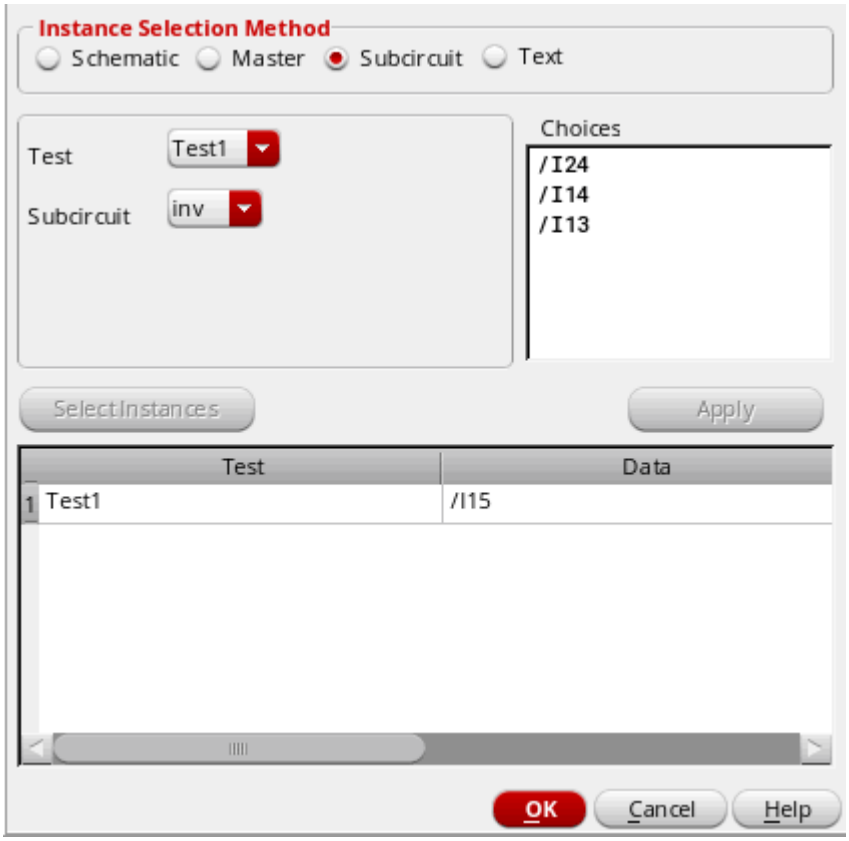
3. In the *Choices* field, select the subcircuit instances you want to include or exclude for applying mismatch variations.

   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.

4. Click *Apply*.


© 2023 Cadence Design Systems, Inc.

The selected instances are added for the test. For example, the following figure shows that the instance `/I15` has been added for the given test.



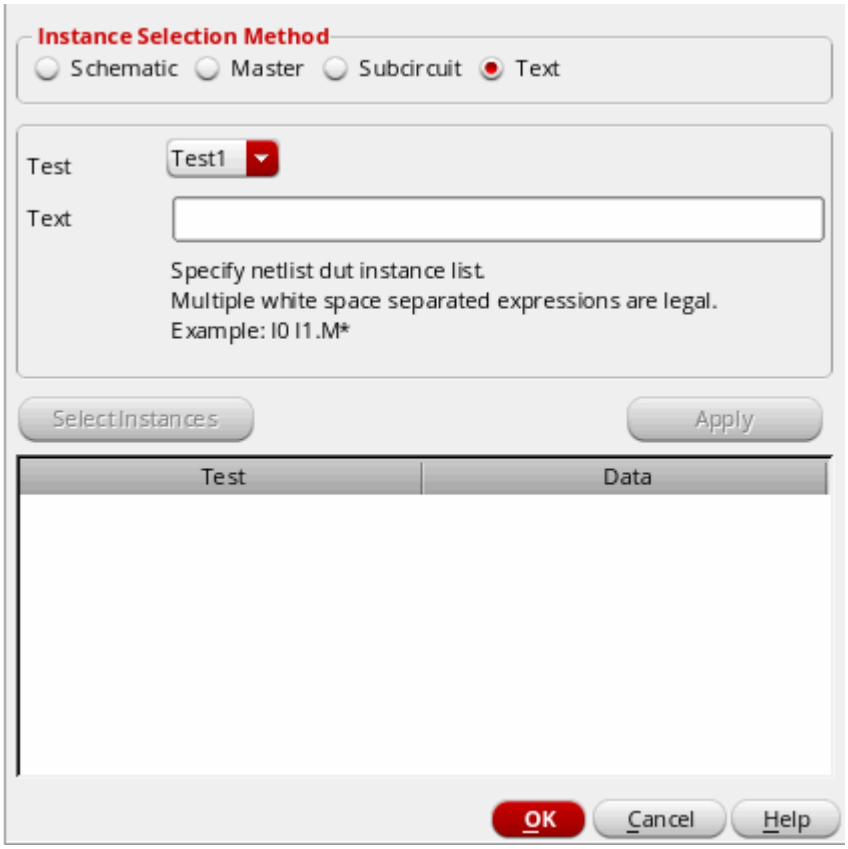

**Note:** By default, the *Type* and *Sch/Master/Subcircuit* columns are not visible. To view these columns, right-click the title bar and select the name of the column to be displayed in this table.


© 2023 Cadence Design Systems, Inc.

## Manually Specifying the Instances for Applying Mismatch Variations

To manually specify the instances you want to include or exclude for applying mismatch variations, do the following:

1. Select the *Text* option.




2. In the *Test* drop-down list, choose the test name.
3. In the *Text* field, enter the names of the instances you want to select. For example, `"I0.I1 I0.I2"`.

   You can also use wildcard matching to specify the instances. For example, `"I0.I1.M8 I0.I2.M* I0.I4"`.

   **Note:** If you use `/` to specify instances in the *Text* field, the tool will convert it into a period. For example, `I0/I1` will be converted as `I0.I1`.

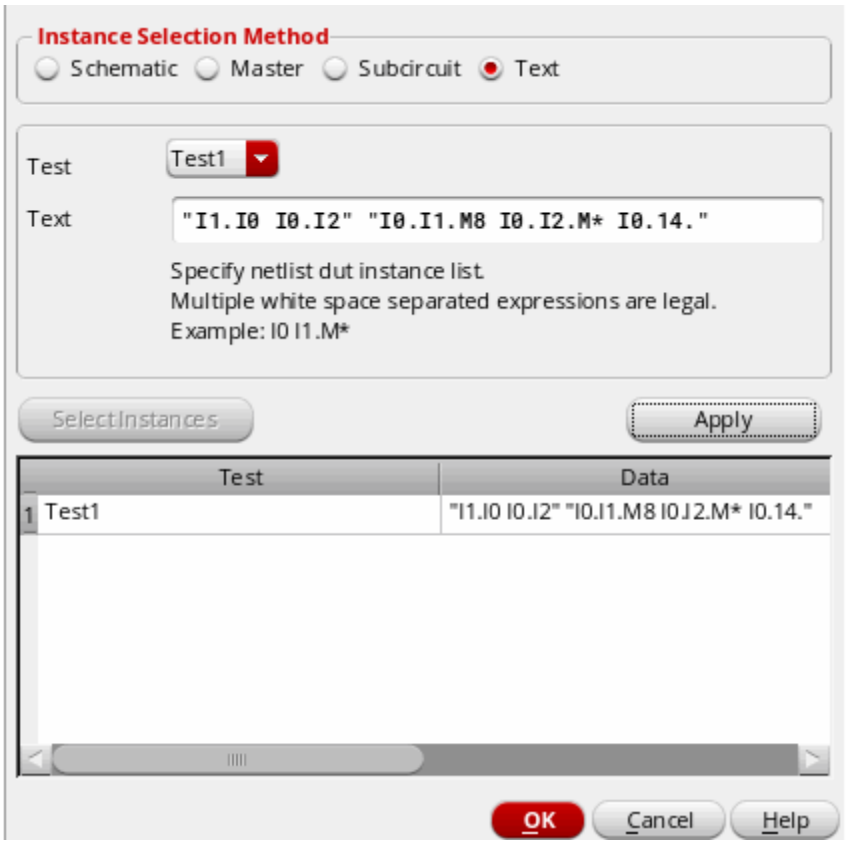

© 2023 Cadence Design Systems, Inc.

*Important*

Wildcard matching is supported only in MMSIM15.1 ISR8 or a later version. In older simulator versions, text selection will work if the wildcard pattern is not used.

4. Click *Apply*.

   The instance selection is added to the test. For example, in the following figure, `"I0.I1 I0.I2"` and `"I0.I1.M8 I0.I2.M* I0.I4"` are added for the `TRAN` and `AC` tests. For the wildcard entry, `I0.I2.M*`, the simulator will consider all the instances prefixed with "`M`" under `I0.I2`.




## Modifying an Existing List of Selected Instances

To modify a list of instances that are already selected for mismatch, do the following:

1. Select a row from the table at the bottom of the Specify Instances for Mismatch form.
2. Modify the list of selected instances in the *Choices* field.


© 2023 Cadence Design Systems, Inc.

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.

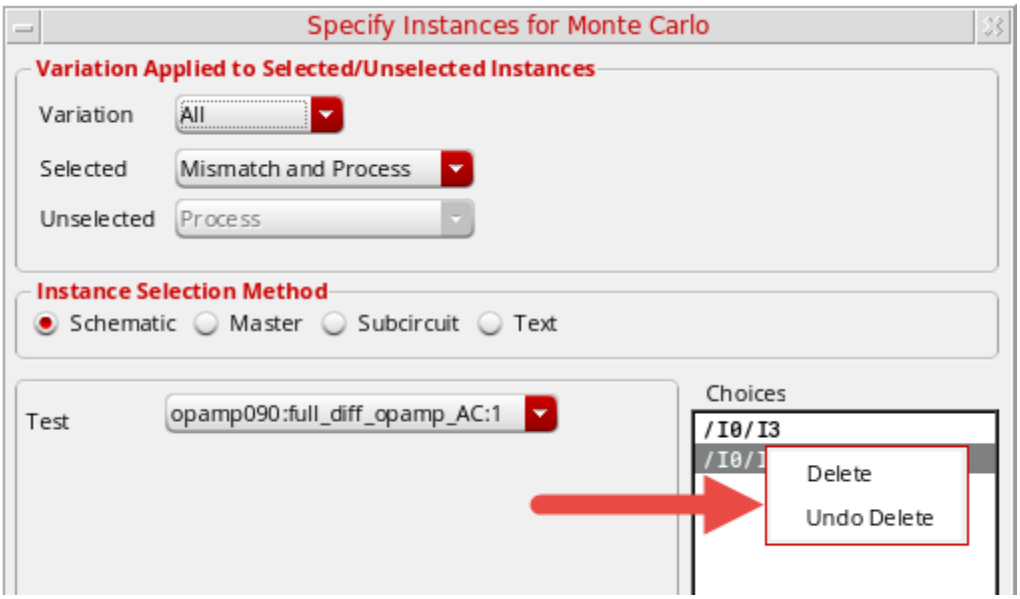
3. Click *Apply*.

## Deleting Instances Selected for Applying Mismatch Variations

You can delete the selected instances in two ways:

- From the Choices list:

  a. Right-click the instance and choose *Delete*.




  b. 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*.

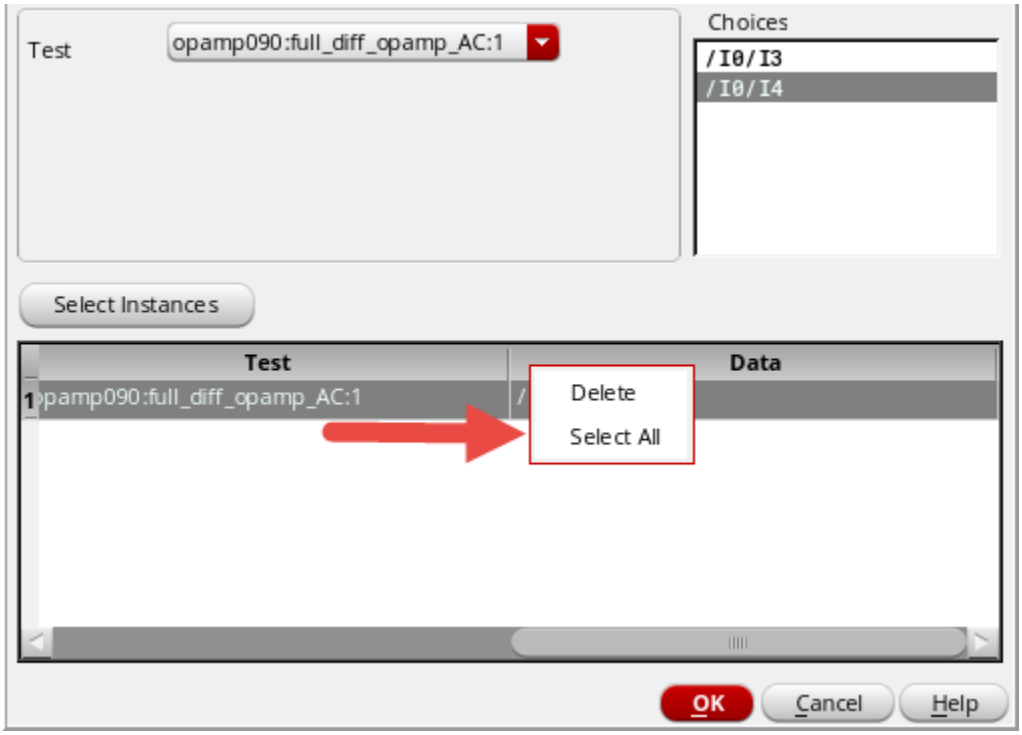
     To undo a delete action, use the *Undo Delete* right-click option.

- From the summary table:

  To delete an instance, do the following:


© 2023 Cadence Design Systems, Inc.

**a.** Right-click the row for the instance and choose *Delete*.




**b.** 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:

**a.** Right-click the row for an instance and choose *Select All*.

**b.** Right-click the row for an instance and choose *Delete*.


© 2023 Cadence Design Systems, Inc.

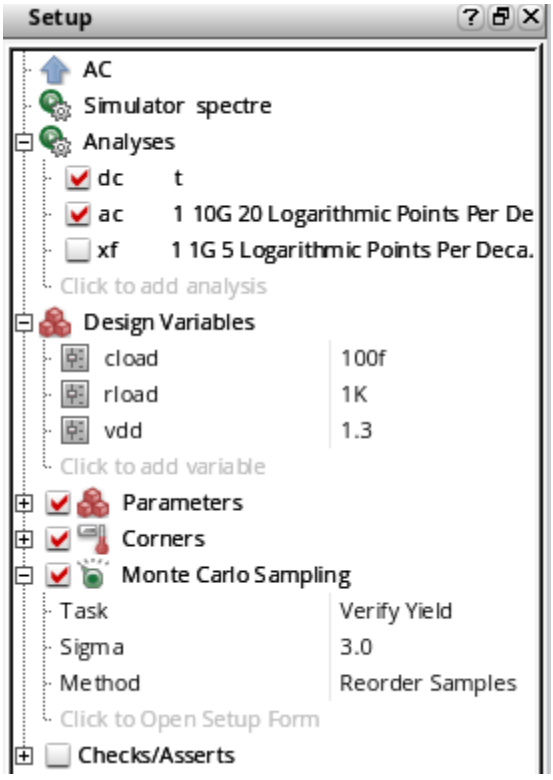
# Running Advanced Methods

When you run a standard Monte Carlo, a single Monte Carlo run history is created and saved in the `ExplorerRun.0` file, which is the default history file used to view saved results in ADE Explorer.

When you have VVO license available, you can run yield verification or other advanced methods using ADE Explorer. For all these methods, the group run history is created and the child histories are saved within the `ExplorerRun.0` file. In addition, the progress bar at the bottom of the ADE Explorer window shows the information and status of each child history run.

To view the child histories, choose *Results – Select*. The Select History to View Results form appears that lists the histories of all the intermediate child runs.

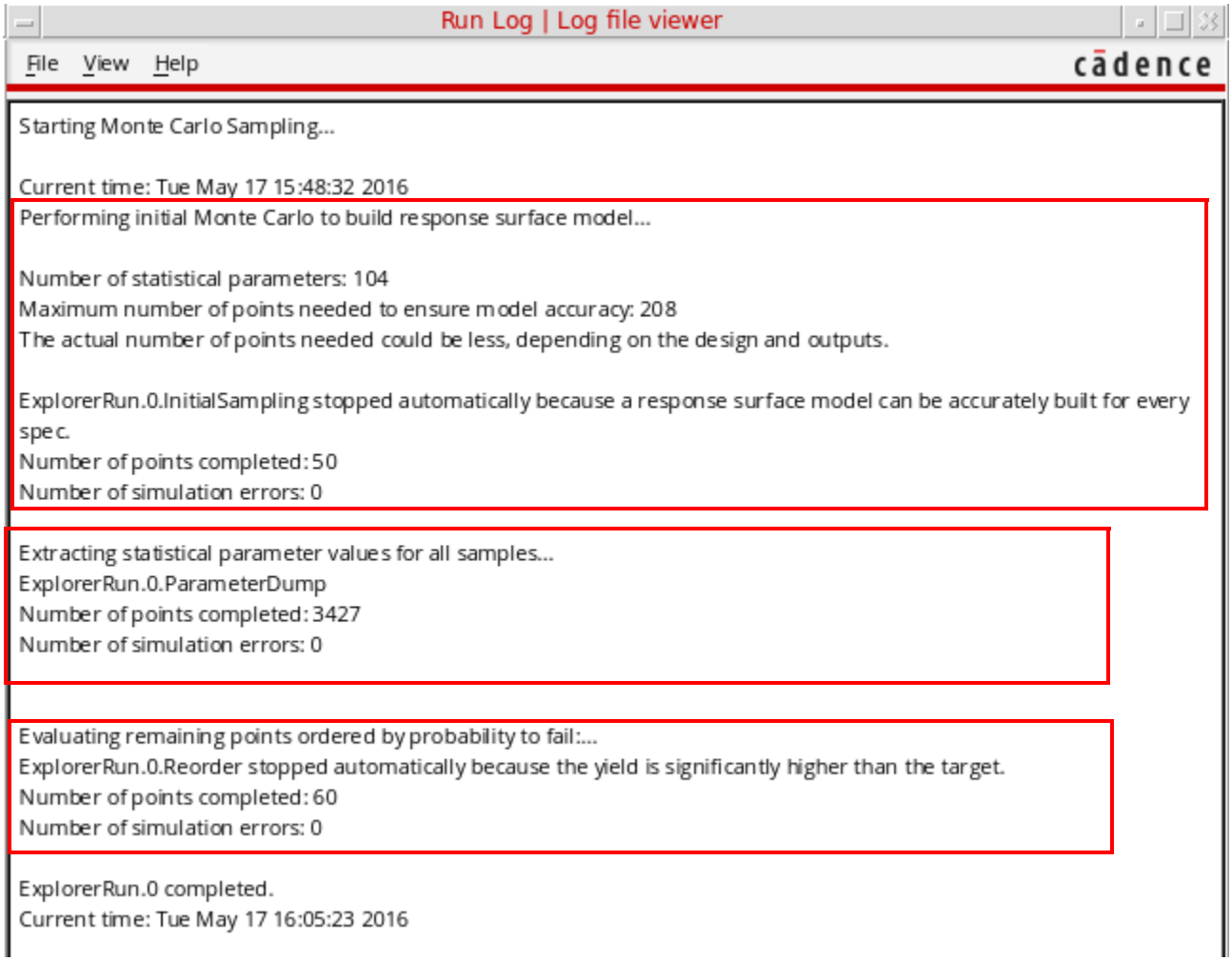
Consider the following example in which you run `Yield Verification - Reorder Samples` method. The Monte Carlo setup has *Task* selected as `Verify Yield`, *Sigma* as `3.0`, and *Method* as `Reorder Samples`.





© 2023 Cadence Design Systems, Inc.

When you run the Monte Carlo simulation, a log file appears that displays information about the tasks performed by the tool during the run.

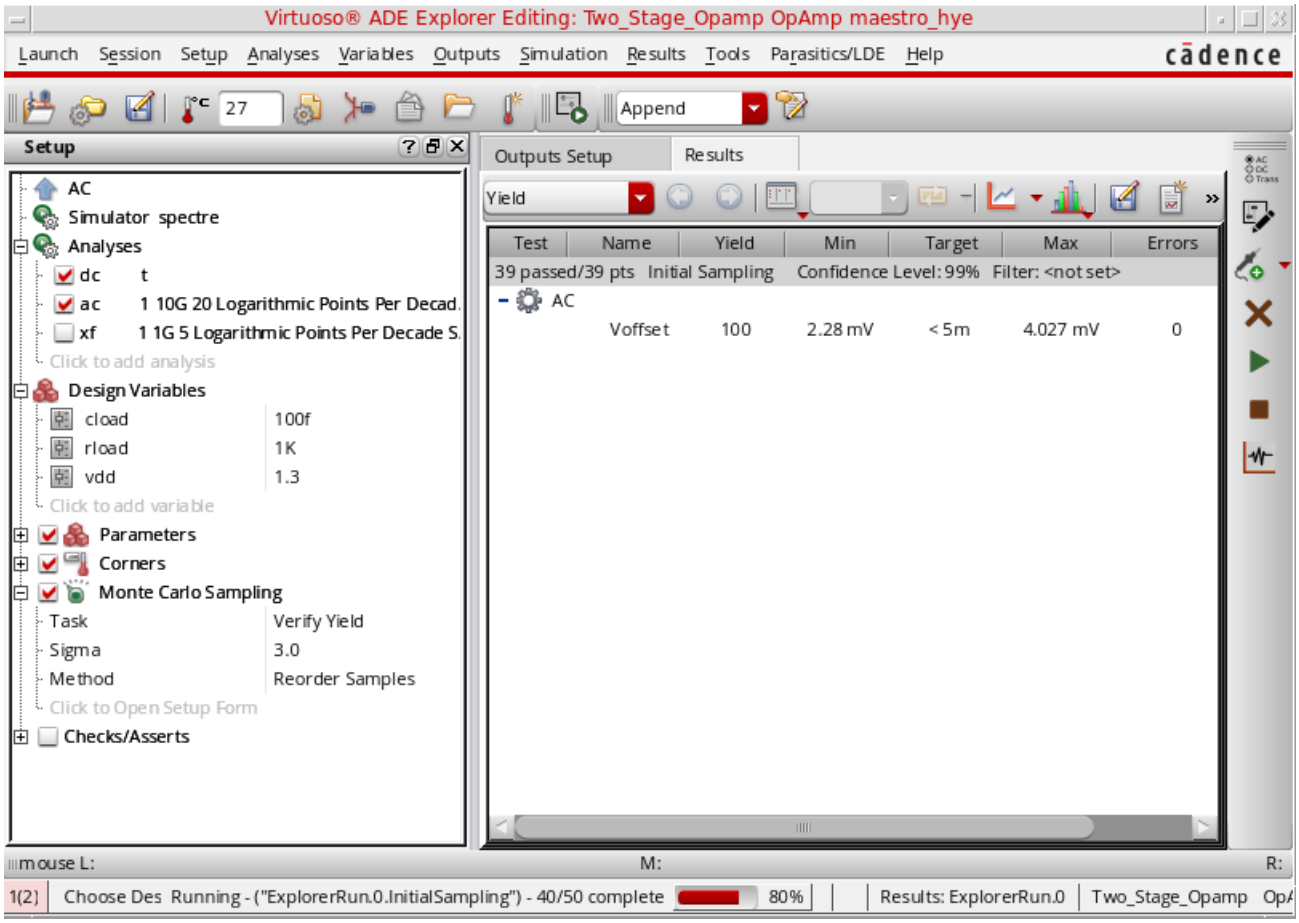



The below steps explain the different intermediate tasks that are performed while the simulation is running:

1. The run begins with an initial Monte Carlo sampling, in which a small number of samples are simulated in the normal order (a minimum of 50 samples) to build the response surface model for each specification. The *Sensitivity Accuracy* method is applied that stops simulating new Monte Carlo samples after enough data is available for accurate modeling. The tool calculates the number of statistical parameters and also the maximum


© 2023 Cadence Design Systems, Inc.

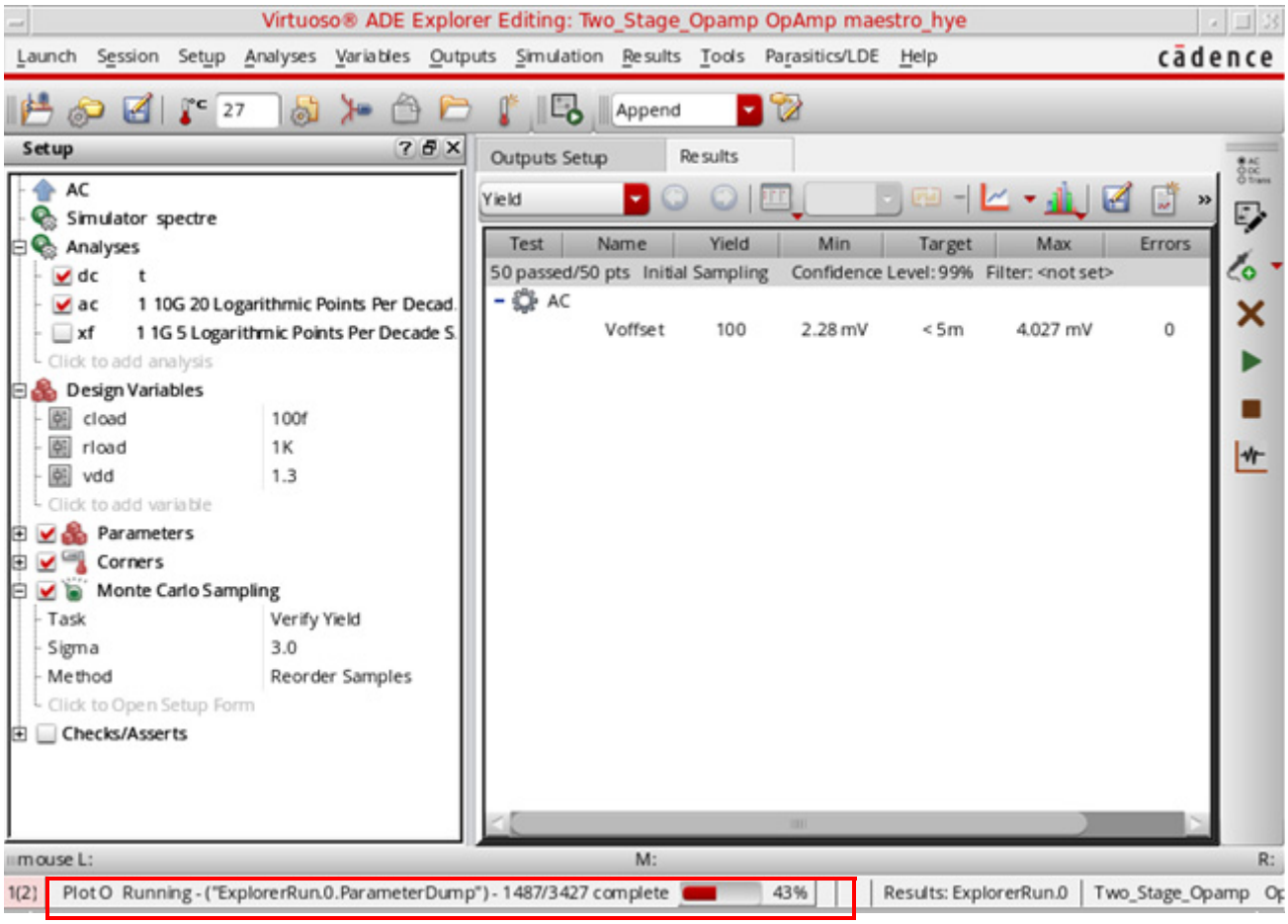
number of points needed to get the model accuracy. However, the number of samples are identified based on the design and outputs, and can be less than the calculated value.




2. After the initial sampling and modeling is complete, the remaining samples are generated but not simulated. The total number of samples required for the run is determined automatically from the target yield and the probability requirements specified in the Monte Carlo form. For each of the remaining samples, the statistical parameter values are generated and the samples are ordered using failure probability modeling. This history of this run is saved in the `ExplorerRun.0InitialSampling` file.


© 2023 Cadence Design Systems, Inc.

The statistical parameters values are extracted for all samples in the `ExplorerRun.0.ParameterDump` file. The progress bar in figure below shows the status of parameter extraction and dumping.




3. Now, the Monte Carlo samples are simulated in the highest probability to fail order. The Monte Carlo run stops automatically in both cases—target yield is met or not met.

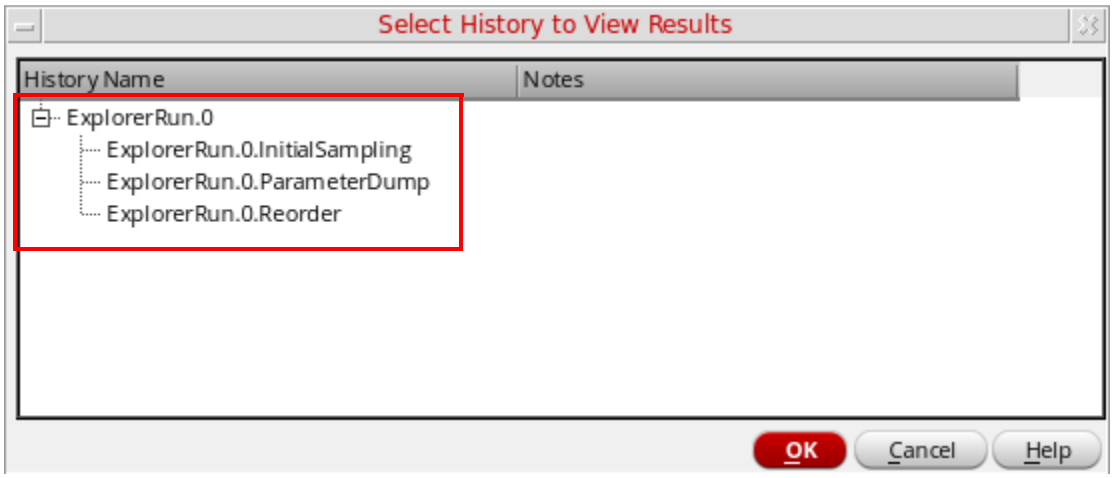
   - When the yield is low, the failed samples are simulated first.
   - When the yield is high, the run stops early when the specifications pass for the samples with high probability to fail. The remaining samples have low probability to fail.

   If the design has low yield, the failed samples are identified early and the simulation run is stopped. On the other hand, if the design has high yield, it can be determined earlier that the yield target is met if the most probable to fail samples pass the specification. If these worst samples pass the specifications, the run stops early.


© 2023 Cadence Design Systems, Inc.

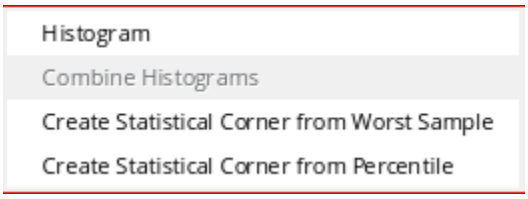
In both the cases, the run stops earlier than the traditional significance test method using a normal random sequence of samples. Significant time saving is observed in high yield cases, since the worst case samples are simulated much earlier in the run.

Now, to view the result histories of this simulation run, choose *Results – Select*. The Select History to View Results form appears, as shown in the figure below.




**Note:** Notice that the run histories for child runs, as mentioned in the steps above, are grouped under the `ExplorerRun.0` file.

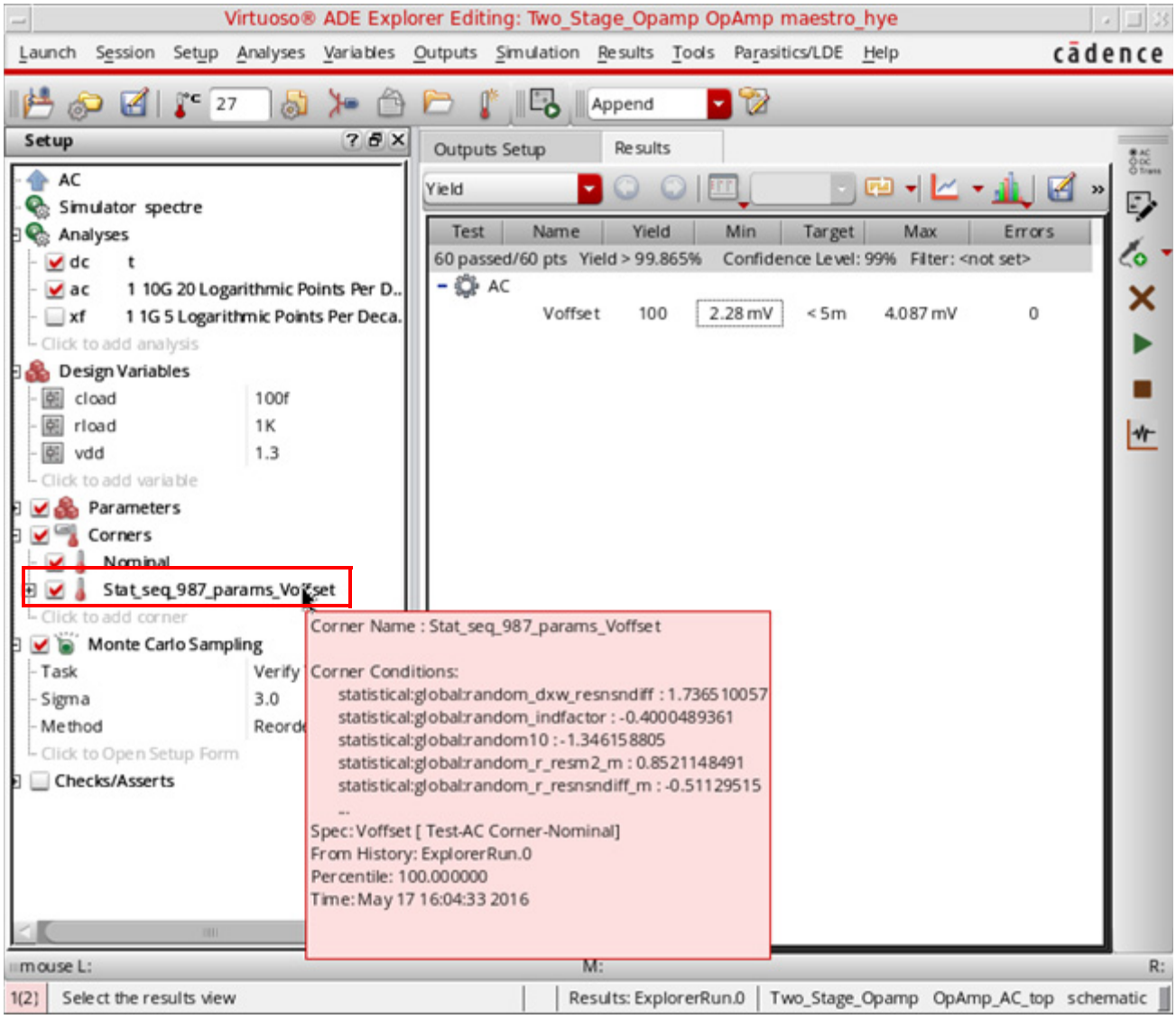
After the run is complete, as a post-processing step, you can also create statistical corners for a specification from the generated results. To do this, right-click a result value and choose Create Statistical Corner from Worst Sample or Create Statistical Corner from Percentile.





© 2023 Cadence Design Systems, Inc.

For example, for the above Monte Carlo setup (Yield Verification - Reorder Samples method), when you create the statistical corner for `Voffset` using Worst Samples, the `Stat_seq_987_params_Voffset` corner is created as shown in the figure below.





© 2023 Cadence Design Systems, Inc.

# 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. You can also configure the values that you display in the Monte Carlo results in the Yield view.

To do this, select the options in the *Configure what is shown in the table* drop-down list available on the *Results* toolbar, as shown in the figure below:

Also see:

- Managing Monte Carlo Results in the Yield View
- Plot/Print versus Iteration
- Plotting Histograms
- Plotting Scatter Plots


© 2023 Cadence Design Systems, Inc.

# 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:

- Expanding and Collapsing Output Data
- Viewing Data for a Specific Confidence Interval
- Filtering Out Data from the Yield View
- Filtering Out Data from the Yield View

## Expanding and Collapsing Output Data

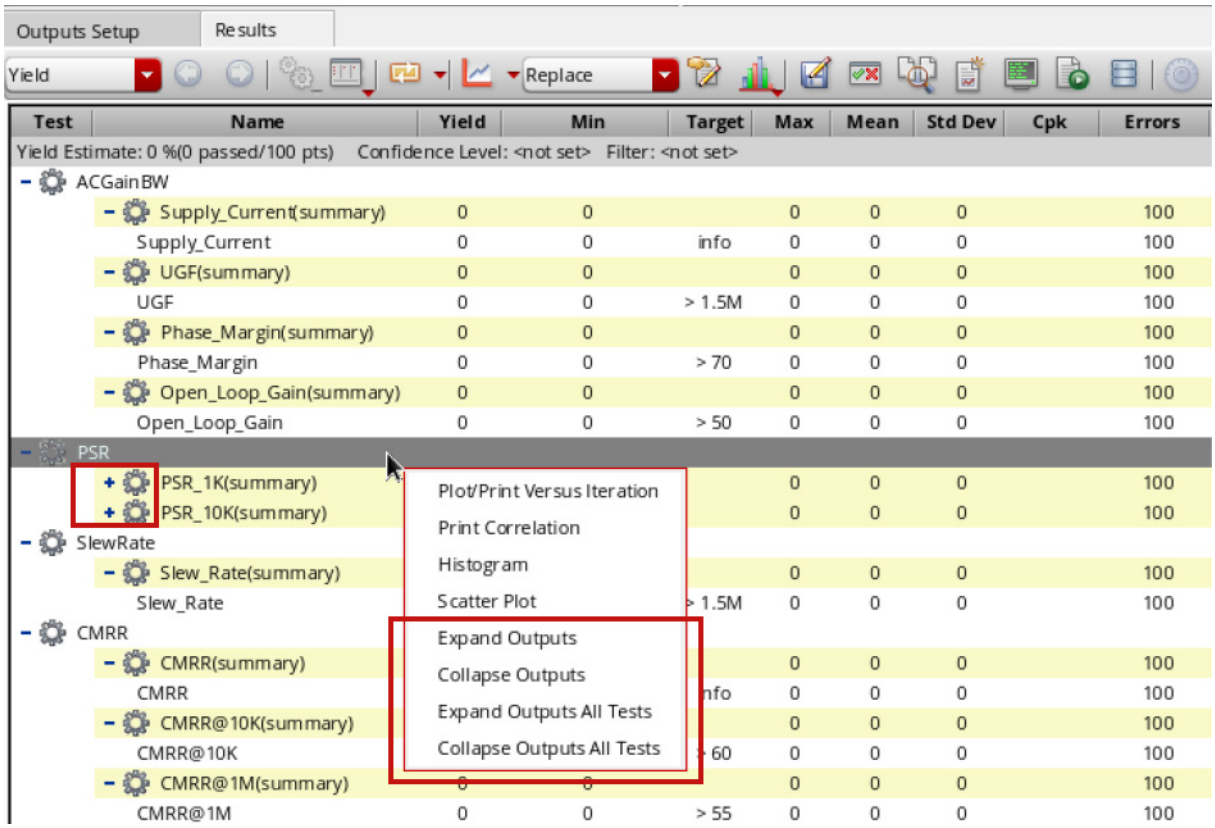
After a Monte Carlo analysis run is complete, you can view the outputs for all tests in an expanded or collapsed form in the *Yield* view on the *Results* tab of the Outputs Setup pane.

To change the view of the outputs of your ADE Explorer tests on the *Results* tab, right-click a testbench and choose the following from the context-sensitive menu: •

- *Expand Outputs to expand the outputs for a specific test*
- *Collapse Outputs to collapse the outputs for a specific test*
- *Expand Outputs All Tests to expand the outputs for all tests*
- *Collapse Outputs All Tests to collapse the outputs for all tests*


© 2023 Cadence Design Systems, Inc.

The following figure illustrates the collapsed outputs for a test in the Yield view on the Results tab:




The *Yield* view displays the number of output rows number specified in the yieldViewMaxOutputsShowExpanded environment variable in expanded form, but when the number of output rows exceeds this limit, the output rows are collapsed by default and you can expand the required rows.

## 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.

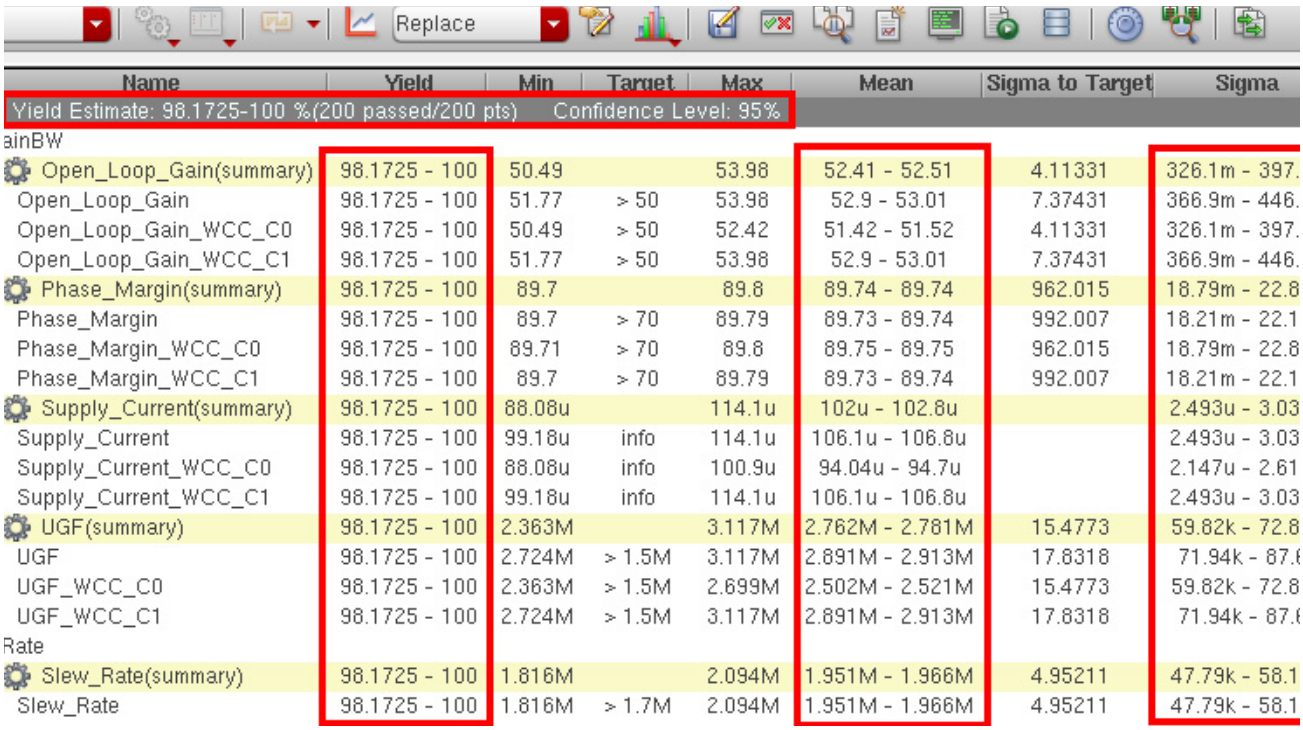

© 2023 Cadence Design Systems, Inc.

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.

Specify a percentage value in the *One-sided Confidence Level* field and click *OK*.

ADE Explorer displays the confidence intervals for the yield, mean, and sigma values, as shown in the following figure.

**Figure 8-1 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 *One-sided Confidence Level* field.


© 2023 Cadence Design Systems, Inc.

## Filtering Out Data from the Yield View

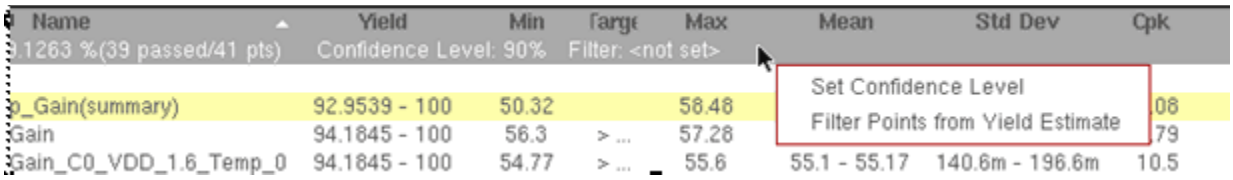
By default, the statistical calculations displayed in the *Yield* view are based on all the points in Monte Carlo results. However, you can filter out and eliminate the points that were either partially complete or they resulted in simulation or evaluation errors so that the yield estimate is calculated only on the points for which simulation was successfully completed.

*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.

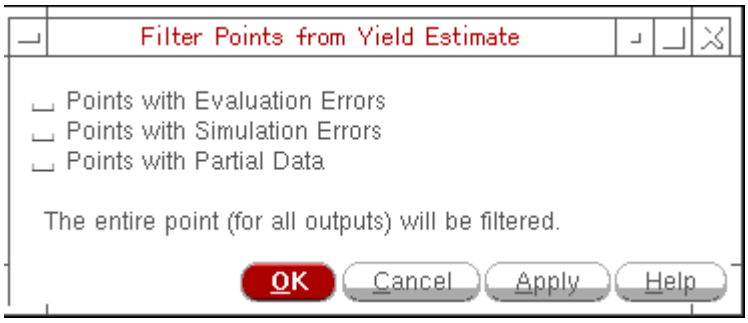
To apply a filter in the *Yield* view, right-click the gray-colored row at the top and choose *Filter Points from Yield Estimate*.

**Figure 8-2 The Context-Sensitive Menu of the title row**




Alternatively, you can right-click the *Errors* column title in the *Yield* view and choose the type of filter to be applied.

The Filter Points from Yield Estimate form is displayed, as shown in the following figure.



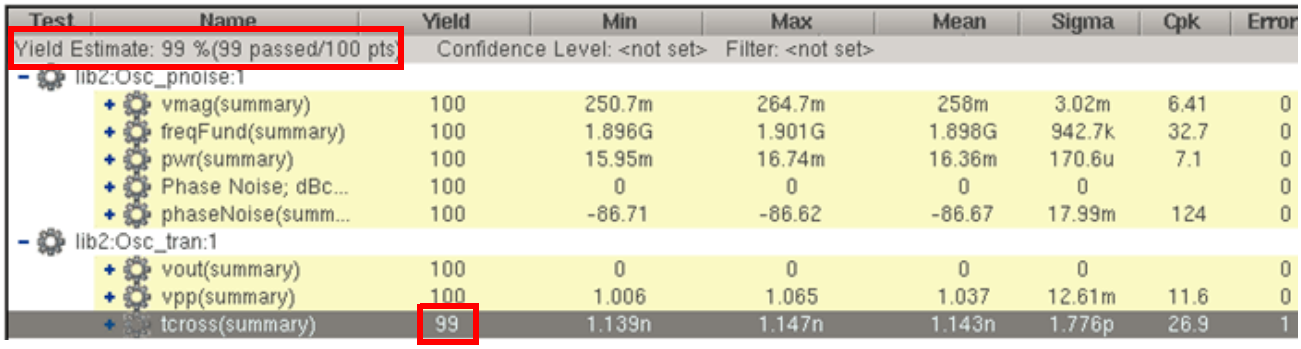

Select a check box to filter the required type of errors and click *OK*.

**Note:** You can select the *Points with Partial Data* check box to filter data from points that were not simulated completely. By default, Virtuoso includes such points, for which only the partial simulation data is available, in yield calculation.


© 2023 Cadence Design Systems, Inc.

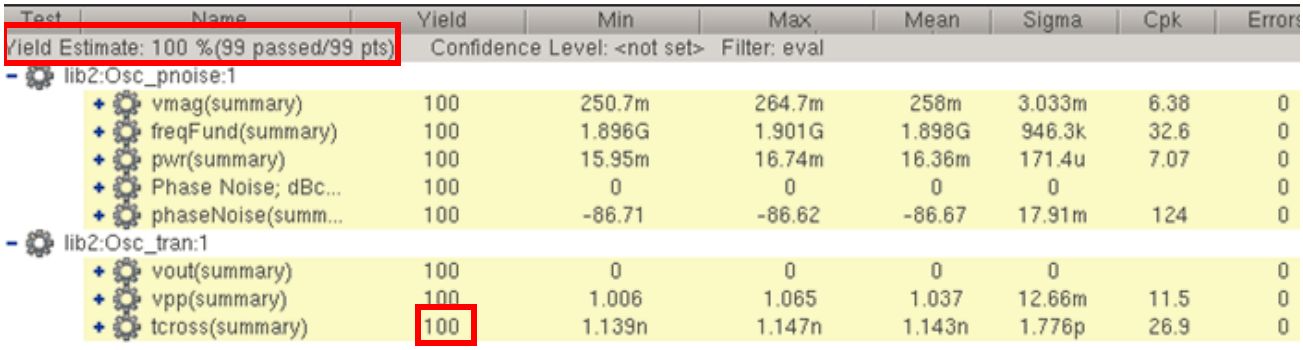
ADE Explorer filters out the incomplete or error points from the *Yield* view. The yield values for specifications and the overall yield estimate are updated. The total number of points considered for yield estimate calculation are also reduced to eliminate the points that were filtered out, as shown in the example figures given below.

**Figure 8-3 Yield View Before Applying the Error Filter**



| Test | Name | Yield | Min | Max | Mean | Sigma | Cpk | Error |
|---|---|---|---|---|---|---|---|---|
| Yield Estimate: 99 %(99 passed/100 pts) | | Confidence Level: <not set> | Filter: <not set> | | | | | |
| lib2:Osc_pnoise:1 | | | | | | | | |
| | vmag(summary) | 100 | 250.7m | 264.7m | 258m | 3.02m | 6.41 | 0 |
| | freqFund(summary) | 100 | 1.896G | 1.901G | 1.898G | 942.7k | 32.7 | 0 |
| | pwr(summary) | 100 | 15.95m | 16.74m | 16.36m | 170.6u | 7.1 | 0 |
| | Phase Noise; dBc... | 100 | 0 | 0 | 0 | 0 | | 0 |
| | phaseNoise(summ... | 100 | -86.71 | -86.62 | -86.67 | 17.99m | 124 | 0 |
| lib2:Osc_tran:1 | | | | | | | | |
| | vout(summary) | 100 | 0 | 0 | 0 | 0 | | 0 |
| | vpp(summary) | 100 | 1.006 | 1.065 | 1.037 | 12.61m | 11.6 | 0 |
| | tcross(summary) | 99 | 1.139n | 1.147n | 1.143n | 1.776p | 26.9 | 1 |

**Figure 8-4 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> | 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.


© 2023 Cadence Design Systems, Inc.

# Generating Plots, Tables, Files, and Reports

After a Monte Carlo run, you can generate plots, tables, files, and reports for your input and output parameters. See the following topics for more information:

- Plot/Print versus Iteration
- Printing Correlation Tables
- Plotting Histograms
- Plotting Scatter Plots
- Exporting Results to a CSV File


© 2023 Cadence Design Systems, Inc.

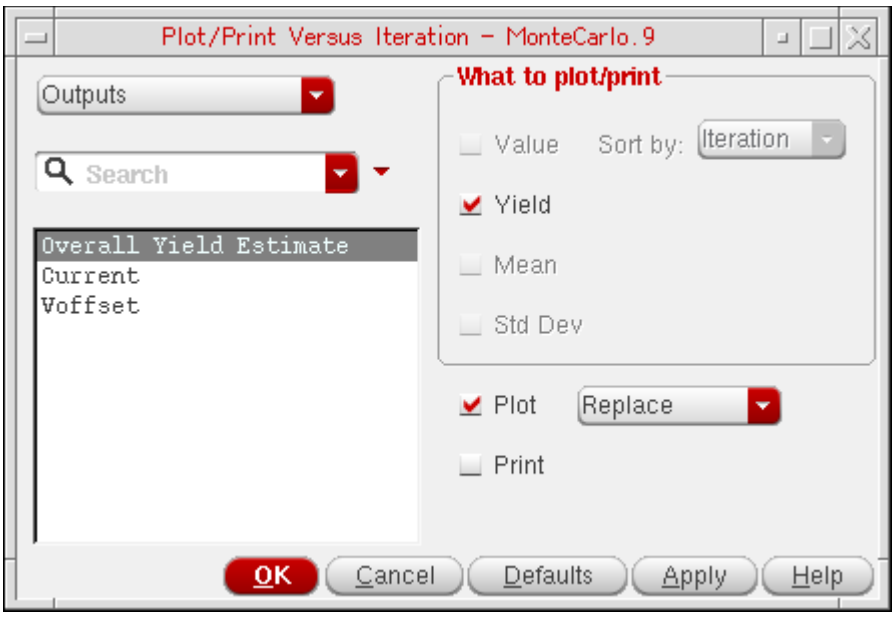
## Plot/Print versus Iteration

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 a test and choose *Plot/Print Versus Iteration.*

   Alternatively, switch to the Yield view, right-click a test name and choose *Plot/Print Versus Iteration*.

   The Plot/Print Versus Iteration form is displayed.




2. From the drop-down list on the top-left corner of the form, 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.


© 2023 Cadence Design Systems, Inc.

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.

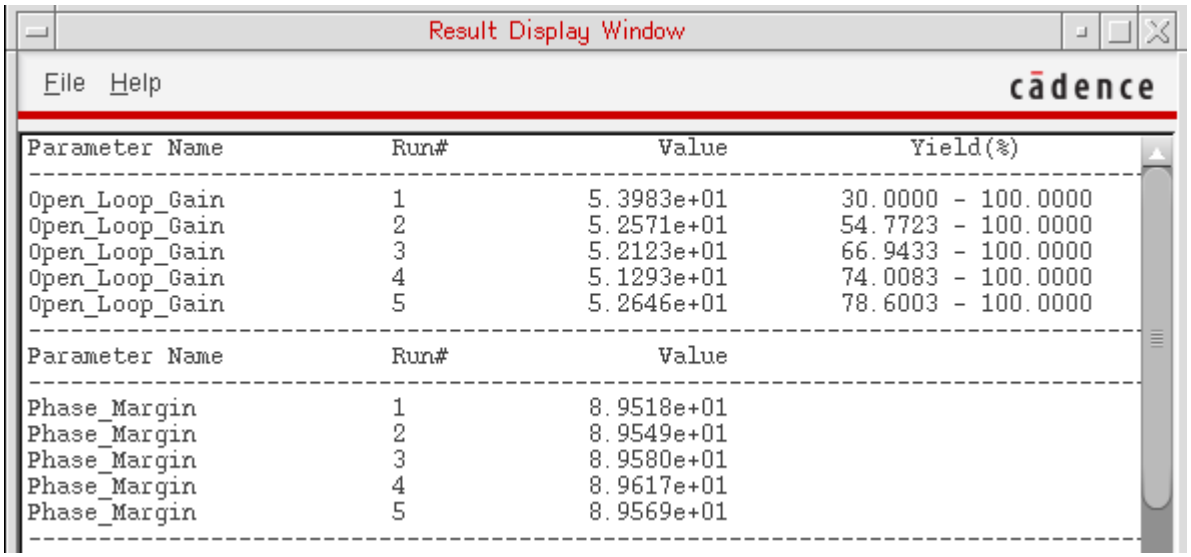
6. Select the *Plot* check box to plot the results.

   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.

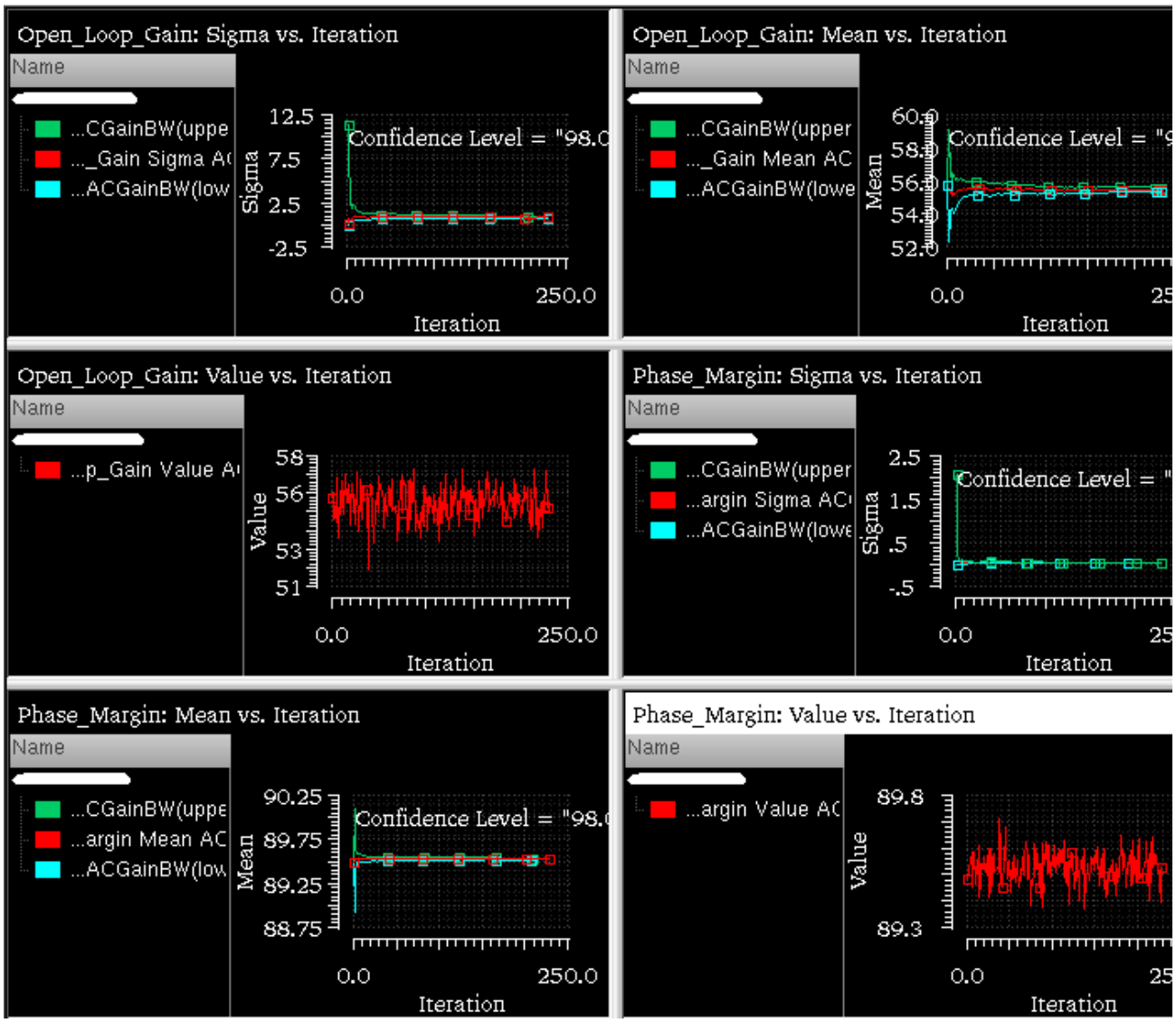
8. Click *Apply* to print or plot as per the specifications given on this form.

   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.


© 2023 Cadence Design Systems, Inc.




**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.


© 2023 Cadence Design Systems, Inc.

***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.

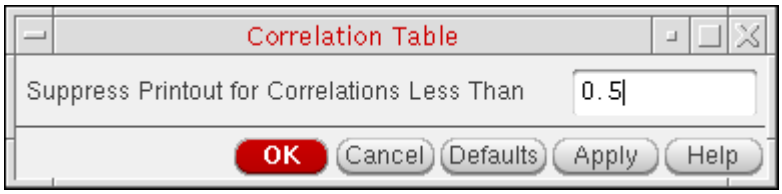
## Printing Correlation Tables

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:

1. Click on the Results tab, select a test and choose *Print Correlation.*

   Alternatively, switch to the Yield view, right-click a test name and choose *Print Correlation*

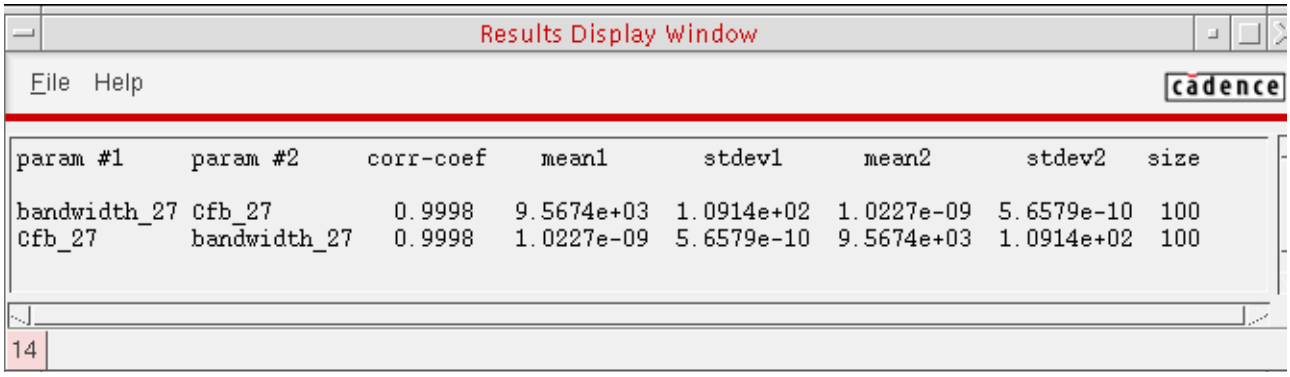
   The Correlation Table form appears.




2. In the *Suppress Printout for Correlations Less Than* field, type a minimum correlation value.
3. Click *OK*.


© 2023 Cadence Design Systems, Inc.

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.


© 2023 Cadence Design Systems, Inc.

## Plotting Histograms

You can plot histograms for the outputs and statistical parameters from Monte Carlo results.

To plot histograms, do the following:

1. Click on the Results tab, select a test and choose *Histogram.*

   Alternatively, switch to the Yield view, 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 list given below that.

2. 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 long, you can use the search field to search for a specific output or parameter name.

3. Specify a numeric value in the *Number of Bins* field.

   The default value is `10`.

   You can change the default value of this field by setting the `histogramBins` environment variable. You can also change the default maximum value for this field from `1000` by setting the `maxHistogramBins` environment variable.

4. 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*


© 2023 Cadence Design Systems, Inc.

5. (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`.

6. (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.
   - *% Markers*: Shows vertical marker lines in the histogram at percentages corresponding to 1, 2, and 3 standard deviations of a normal distribution. If the distribution of the output is normal, these lines will be at the same position as the sigma lines. You can display the marker lines to get an idea about the distribution of points.
   - *Std Dev Lines*: Shows the standard deviation lines in the graph indicating the mean, mean – standard deviation, and mean + standard deviation values.
   - *Show Points*: Shows points on the histogram.

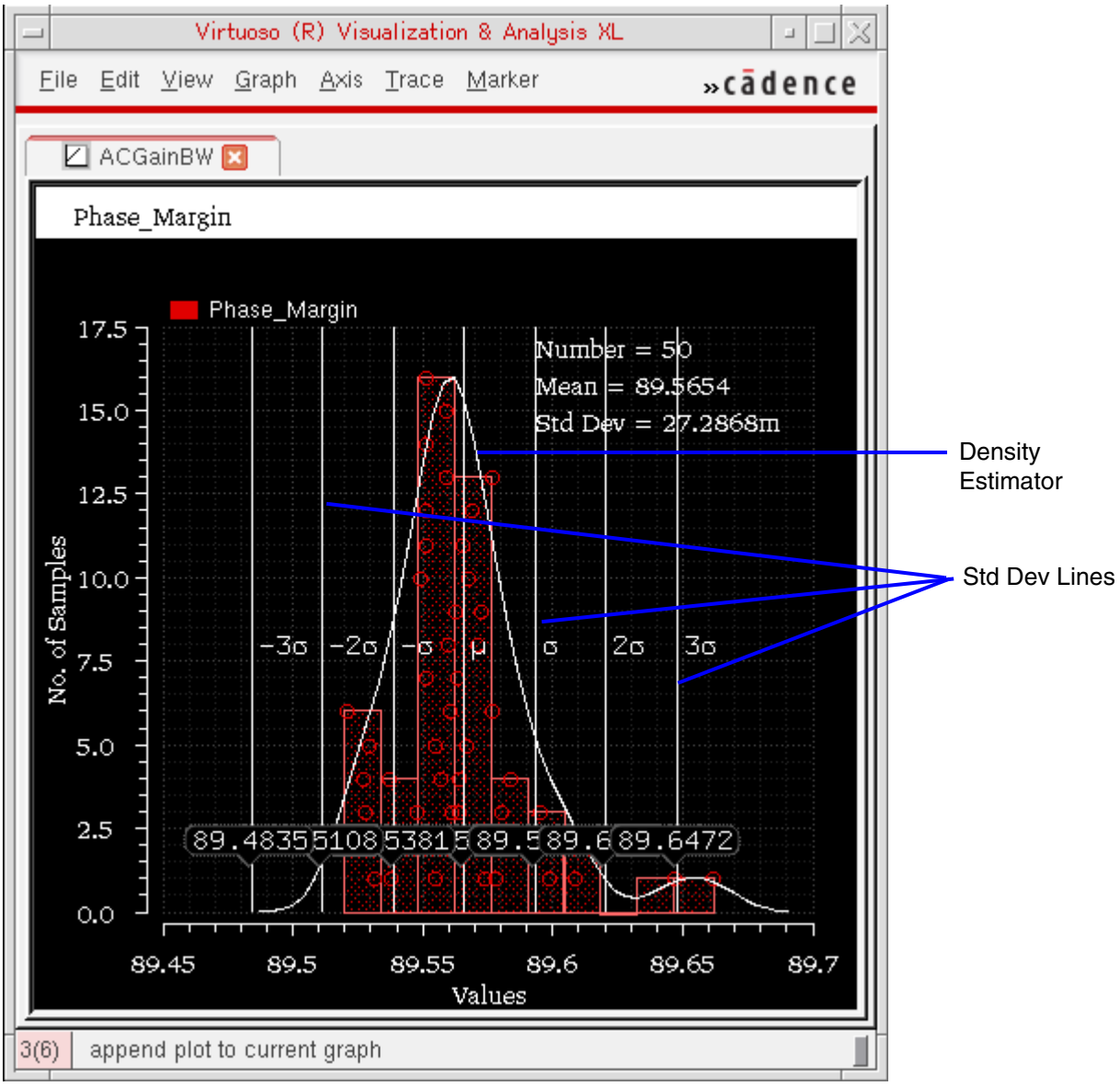
7. (Optional) Select the *Normal Quantile Plots* check box to plot the quantile plots (Q-Q plots).

8. If you have selected multiple corners for a histogram, click *Combine*. For more details, refer to Plotting Combined Histograms.

9. Click *OK*.


© 2023 Cadence Design Systems, Inc.

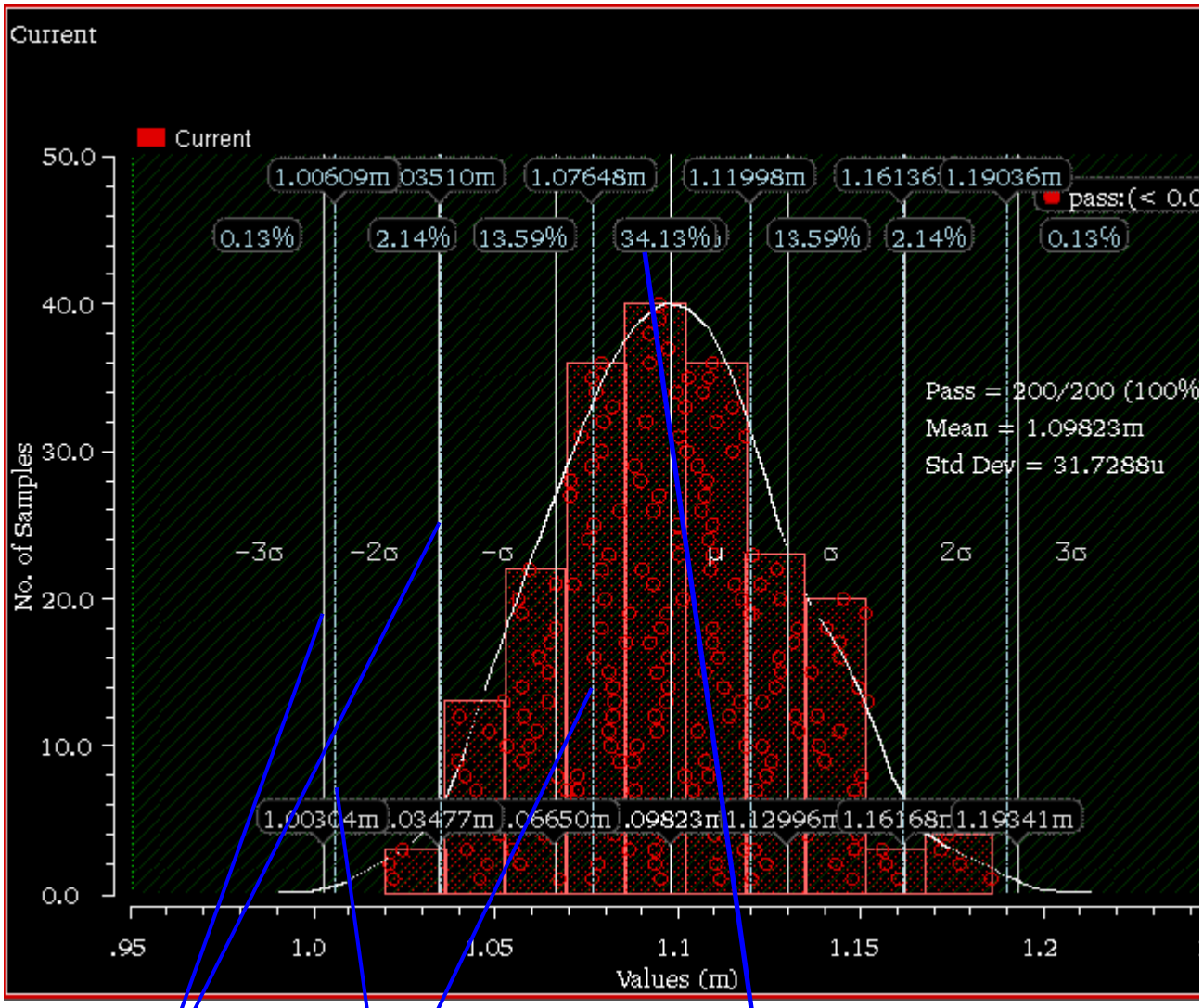
The histogram plot appears in the Virtuoso Visualization and Analysis XL window, as shown below.




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.


© 2023 Cadence Design Systems, Inc.

The example histogram given below shows how % markers are displayed.




Sigma lines

% Markers

The label indicates that 34.13% of the total number of points fall between the mean value and 1.11998m

The vertical marker lines are placed at percentages corresponding to 1, 2, and 3 standard deviations of a normal distribution. The following intervals are bound by vertical lines and labeled, where, a label between two marker lines displays the percentage of points that fall in the given range:

- 0.13% (-infinite to -3 sigma normal distribution)


© 2023 Cadence Design Systems, Inc.

- 2.14% (-3 sigma to -2 sigma normal distribution)
- 13.59% (-2 sigma to -1 sigma normal distribution)
- 34.13% (-1 sigma to mean normal distribution)
- 34.13% (mean to 1 sigma normal distribution)
- 13.59% (1 sigma to 2 sigma normal distribution)
- 2.14% (2 sigma to 3 sigma normal distribution)
- 0.13% (3 sigma to infinite normal distribution)

You can display the marker lines to get an idea about the distribution of points. If the distribution of the output is normal, these lines will be at the same position as the sigma lines.

### *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 these points to cross-probe results on the Results tab. For more details, refer to Cross-Probing Results from Histogram Plots.
- When you choose to display the normal quantile plot, ADE Explorer 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 The Mismatch Contribution Table.
- You can set the histogram options using the Plotting/Printing Options form.
- You can set the default values for the histogram options using the following environment variables:
  - `histogramType`

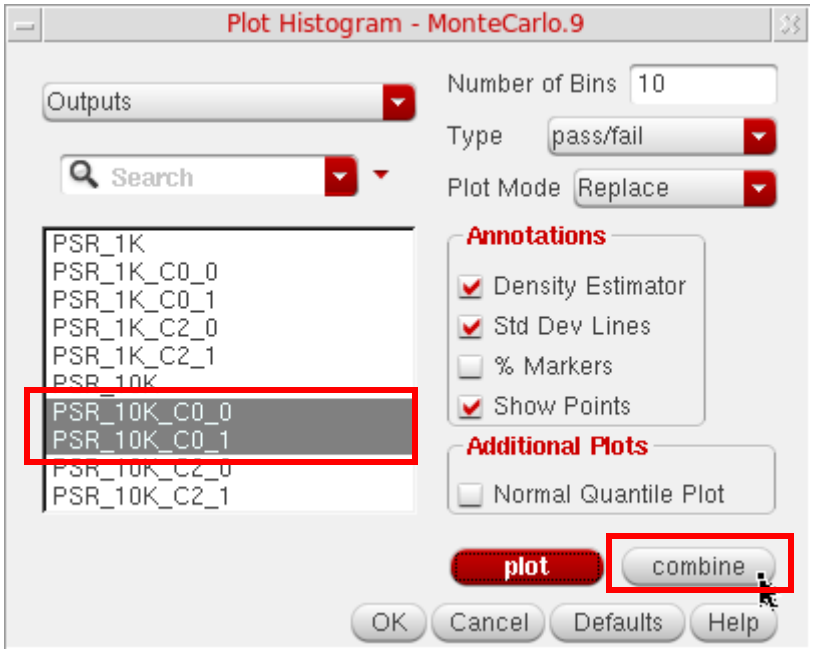

© 2023 Cadence Design Systems, Inc.

- `histogramBins`
- `showHistogramDensity`
- `showHistogramDeviation`
- `showHistogramPercentMarkers`
- `showHistogramPoints`
- `histogramQQPlot`
- `maxHistogramBins`

### Plotting Combined Histograms

When the results of a Monte Carlo simulation contain outputs for multiple corners, you can plot a combined histogram for the outputs of more than one or all the corners. You can plot a combined histogram in any one of the following two ways:

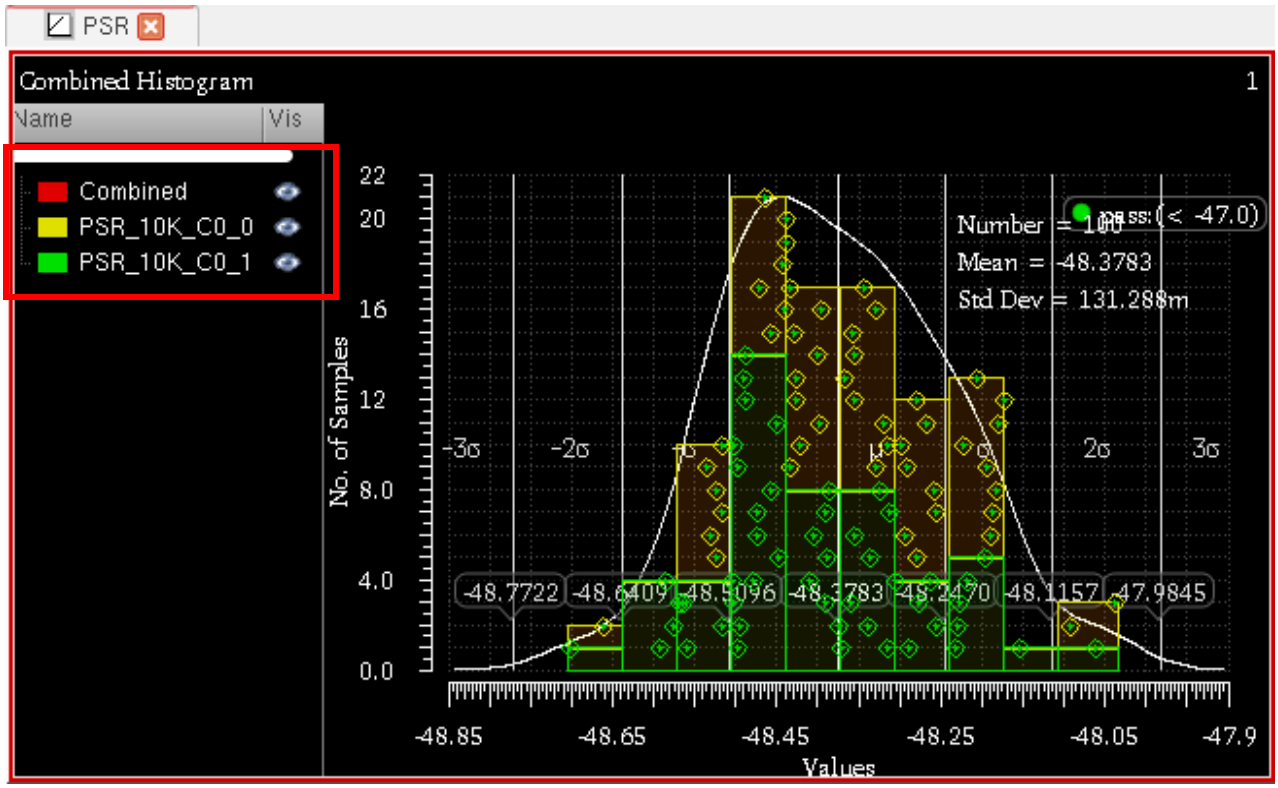
- On the Plot Histogram form, select the names of outputs for multiple corners and click *Combine*.




- In the *Yield* view of the Monte Carlo results, press the `Ctrl` key and select the outputs for multiple corners. Right-click and choose *Combine Histograms*.


© 2023 Cadence Design Systems, Inc.

The Virtuoso Visualization and Analysis XL window is opened displaying a combined histogram for all the selected outputs as well as an individual histogram for each output.



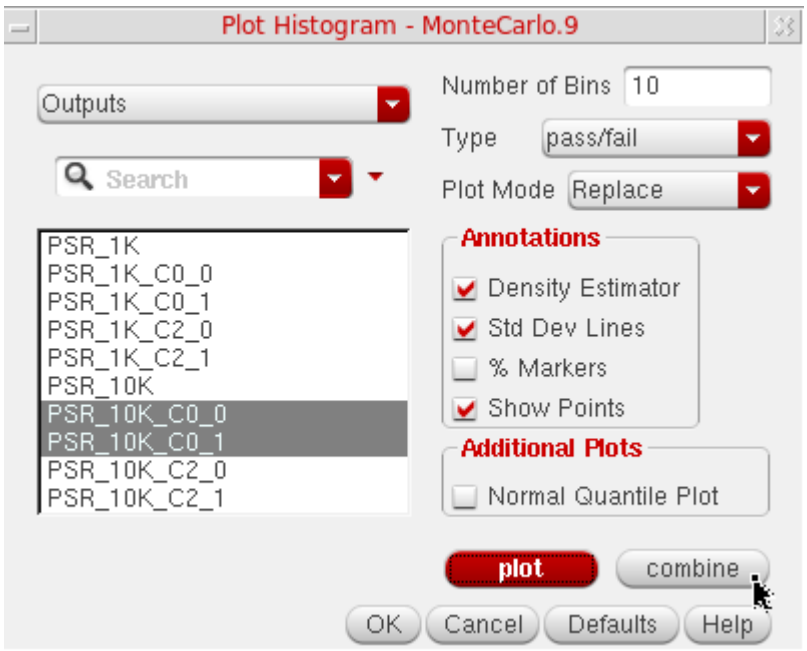

By default, all the histograms are visible. However, you can use the visibility (Vis) marker in the graph legend to control the display of an individual histogram.

### *Important Points to Note*

- A combined histogram shows the combined values for all the corners used to plot the histogram. For example, it shows the mean or standard deviation value calculated over all the included corners.


© 2023 Cadence Design Systems, Inc.

- You can move a leaf out of a combined histogram into a new window, subwindow, or strip, as shown below.



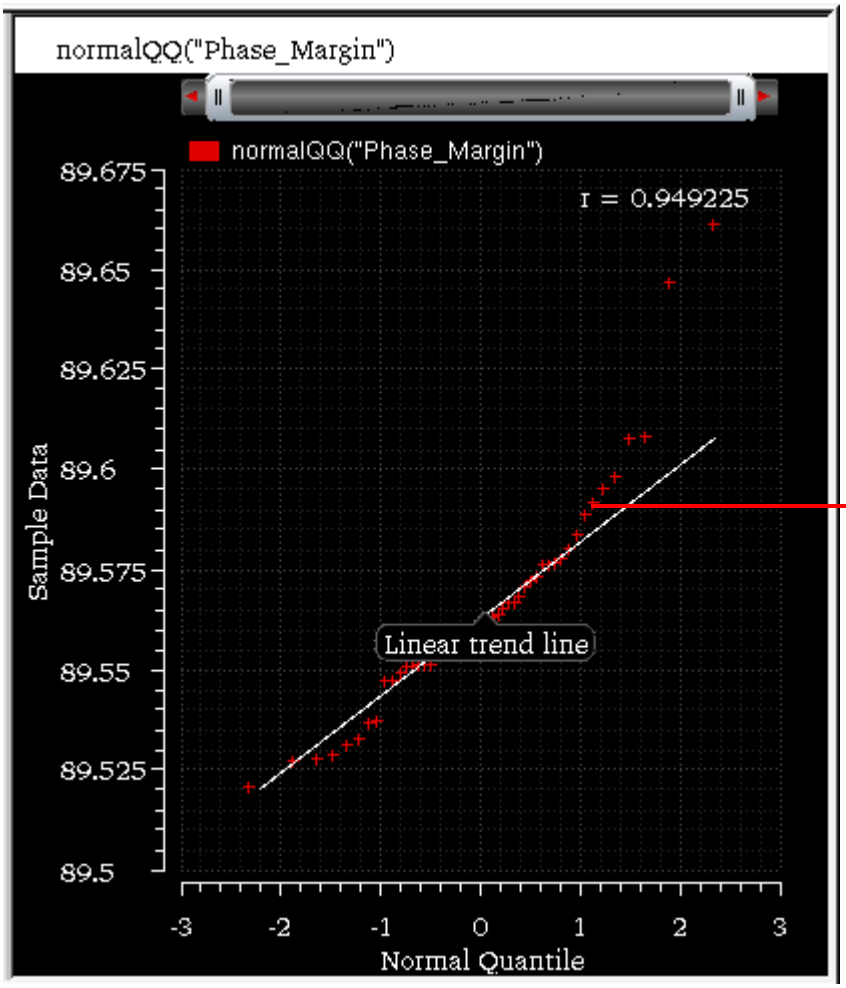

- You can right-click the name of any output in the graph legend and click *Delete* to remove the histogram for that output. The combined histogram is also redrawn to show the updated calculation.
- After plotting a combined histogram, you can drag and drop a histogram for another corner, but of the same output, on it to combine it with the existing one. The data from the newly added histogram would be added to the combined histogram.
- You can cross-select data points from an individual leaf of a combined histogram into the *Detail* view of results.
- You cannot merge multiple combined histograms into a single graph.
- You cannot turn off the visibility of combined histograms.
- It is not possible to plot histograms for individual corners and then combine them.
- If you select the *Normal Quantile Plot* check box, a sub window containing a combined normal quantile plot (Q-Q plot) is created. It contains all the points from the selected corners merged into a single Q-Q plot. You can change the visibility status of each leaf in a combined Q-Q plot and also cross-select data points from an individual leaf into the *Detail* view of results. However, you cannot move or delete individual leaves from this plot.


© 2023 Cadence Design Systems, Inc.

### Normal Quantile Plots

A normal quantile plot (Q-Q plot) helps in comparing the result values for an output with the values predicted for a standard normal distribution. Based on the threshold value, a Q-Q plot may be in the following three states:

- Normal
- Near Normal
- Non Normal




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

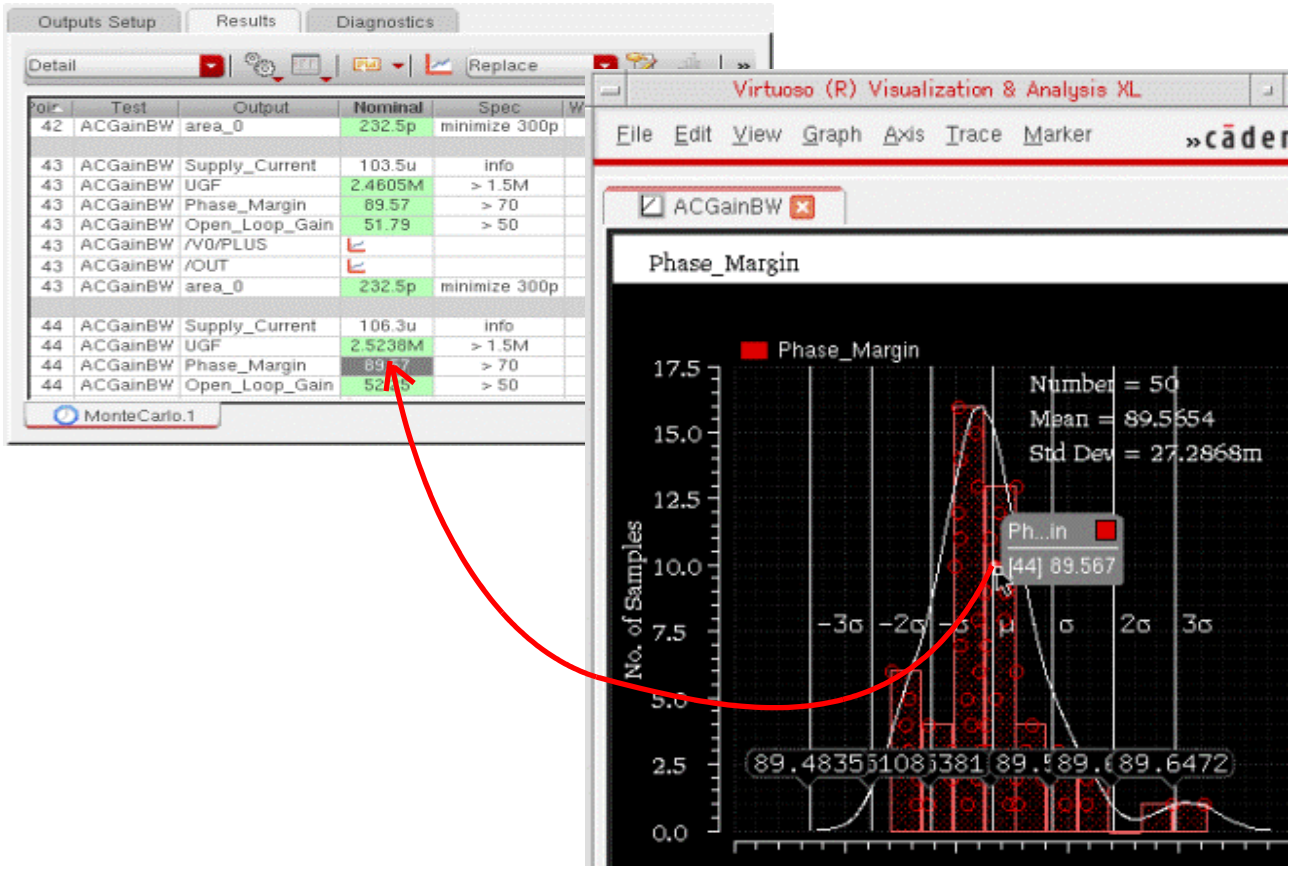

© 2023 Cadence Design Systems, Inc.

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 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 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.

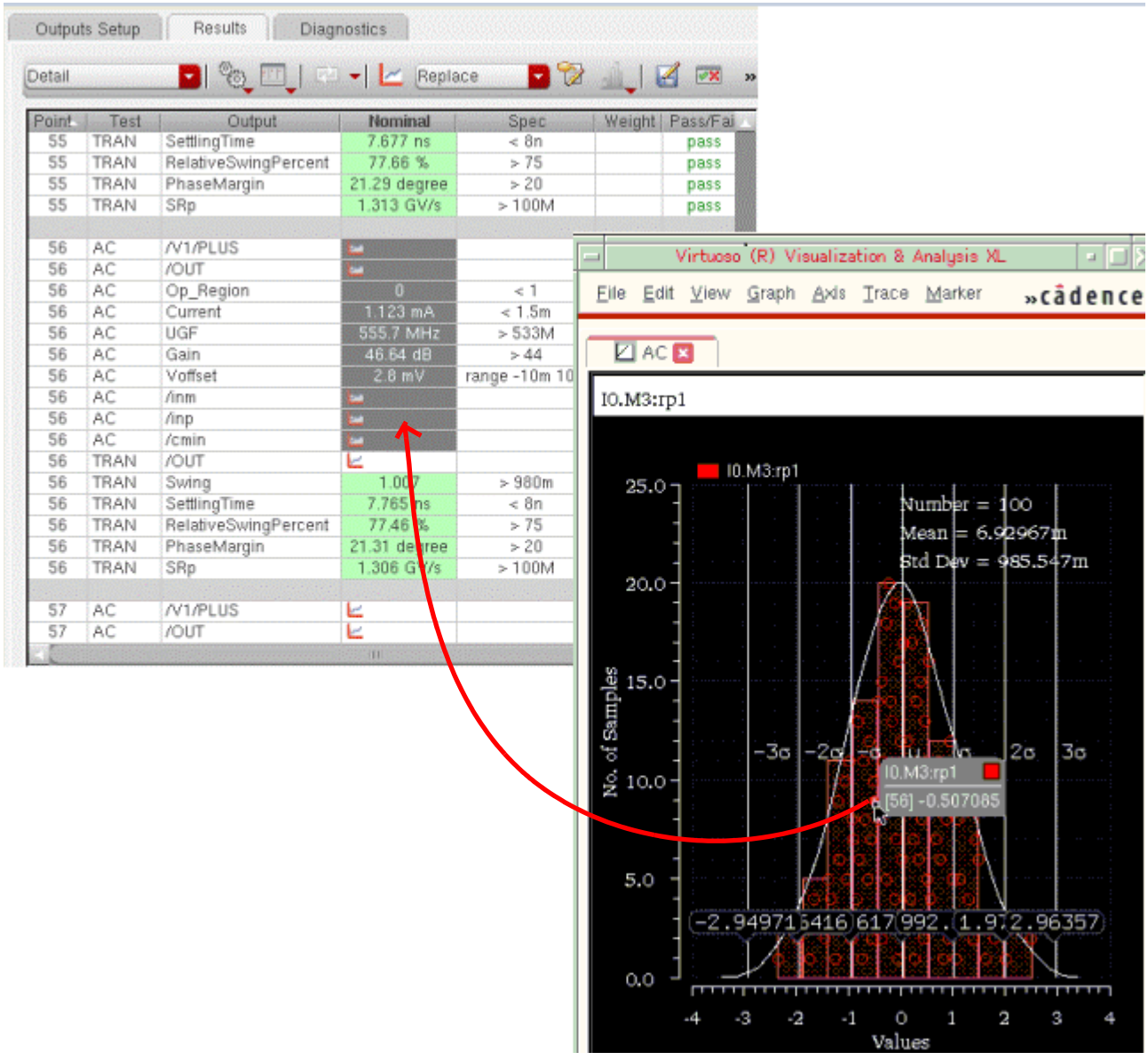




© 2023 Cadence Design Systems, Inc.

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.

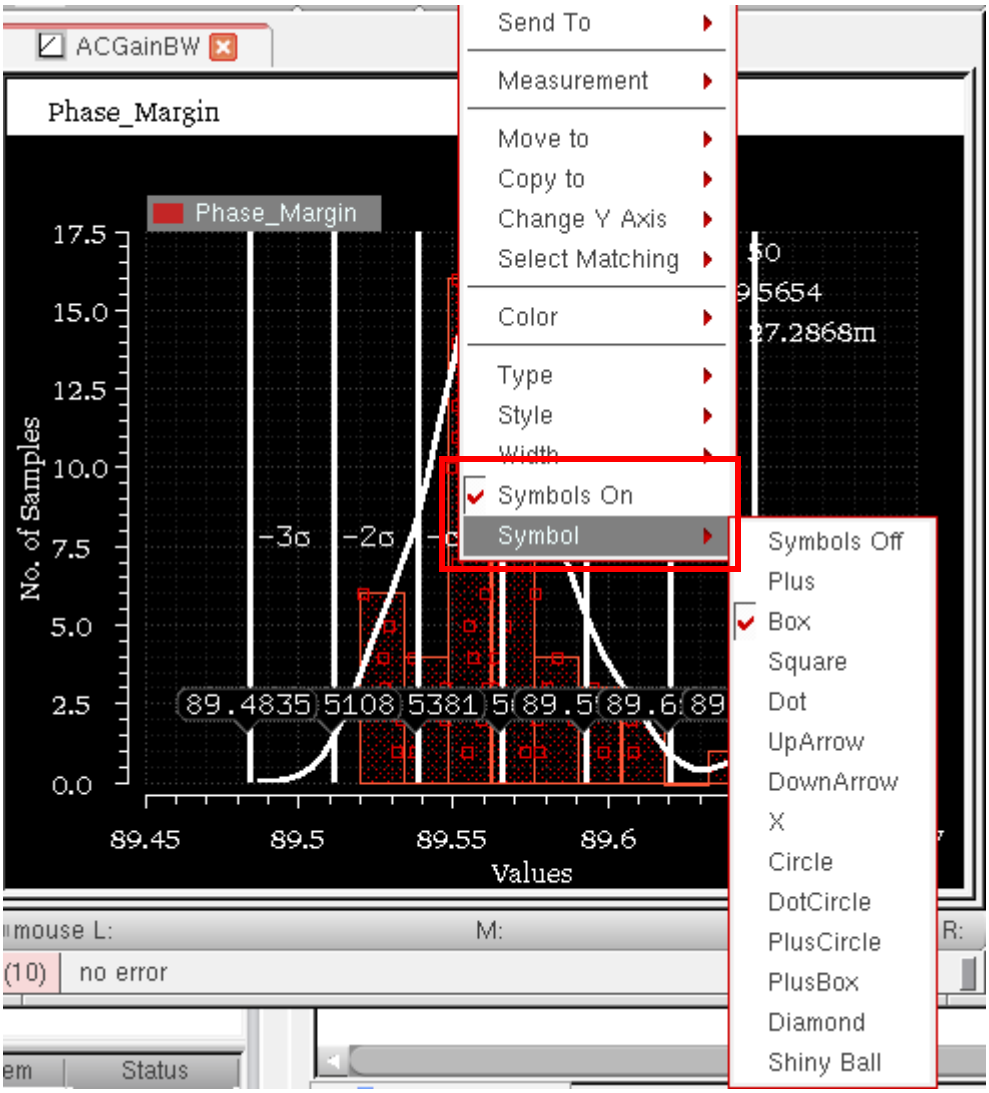



If you do not need to cross-probe results from histogram, clear the *Show Points* check box on the Plot Histograms form or on the Plotting/Printing Options form. You can also set the `showHistogramPoints` environment variable to `nil`. This fills the bars with solid red color


© 2023 Cadence Design Systems, Inc.

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.





© 2023 Cadence Design Systems, Inc.

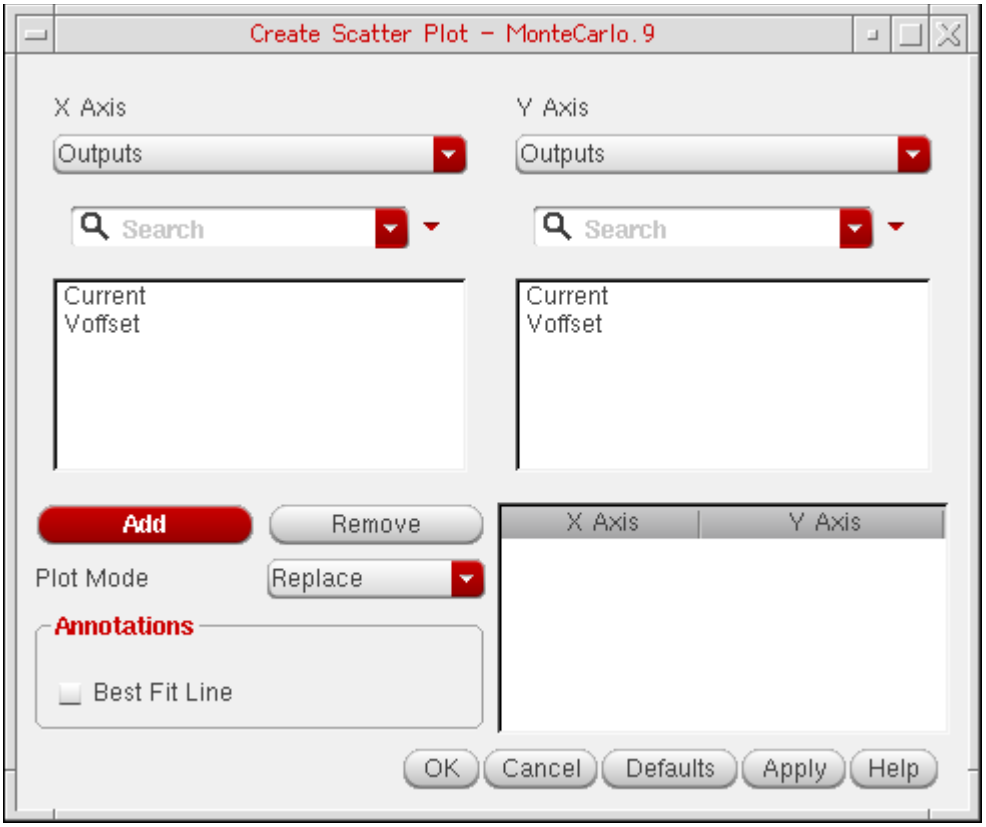
## Plotting Scatter Plots

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. Click on the Results tab, select a test and choose *Scatter Plot.*

   Alternatively, switch to the Yield view, right-click a test name and choose *Scatter Plot.*

   The Create Scatter Plot form is displayed.




   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.

2. To make an output pair, select an output from each one of the *X-axis Data* and *Y-axis Data* list.

3. Click *Add*.

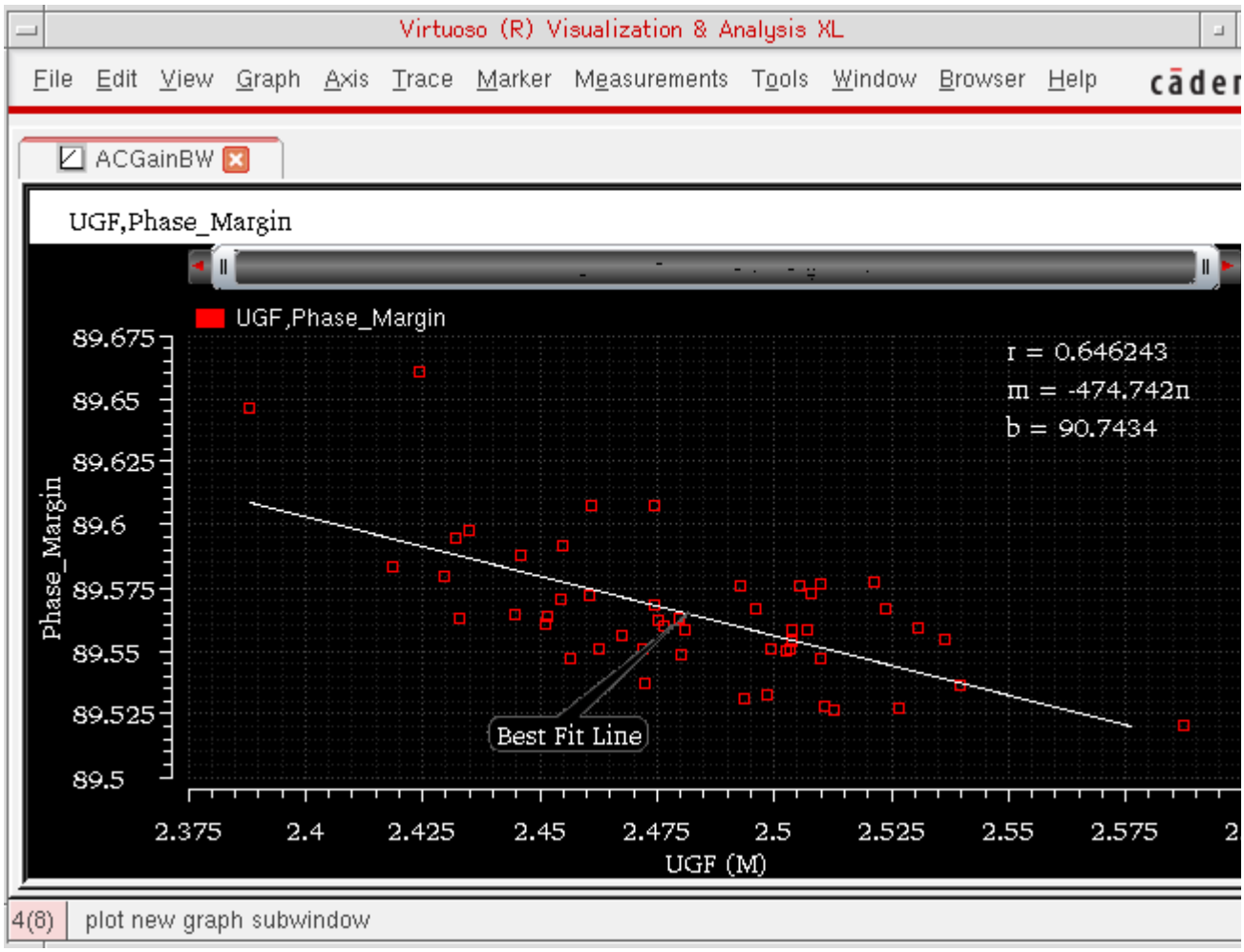
   Similarly, you can select more outputs or parameters. The selected pairs appears in the list box shown in the bottom right corner of the form.


© 2023 Cadence Design Systems, Inc.

4. (Optional) Select a plotting mode from the *Plot Mode* list. The default value is `Replace`.
5. (Optional) Click *Best Fit Line* to draw a least-squares-fit line for each parameter pair.
6. 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`.


© 2023 Cadence Design Systems, Inc.

- `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`.

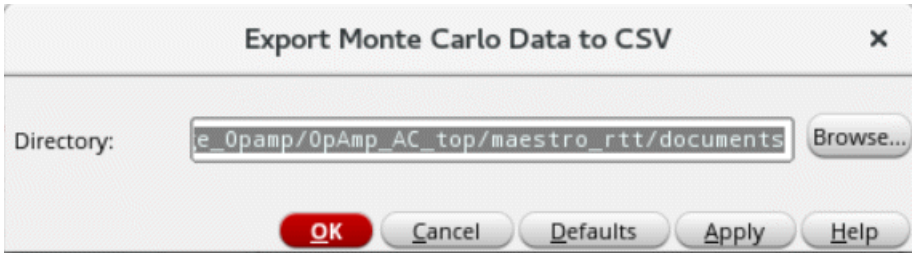

© 2023 Cadence Design Systems, Inc.

### Exporting Results to a CSV File

To export Monte Carlo results to a CSV file:

1. Click on the Results tab, select a test and choose *Export Data to CSV.*

   The Export Monte Carlo Data to CSV form is displayed.




2. Click *Browse* to open the Select Export Directory form or manually specify the path to the export directory.
3. Click *OK*.

# Viewing Statistical Parameters for Monte Carlo Samples

After running a Monte Carlo analysis, you can switch to the Detail view on the Results tab 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, select *Detail*.

   The Detail view is displayed.

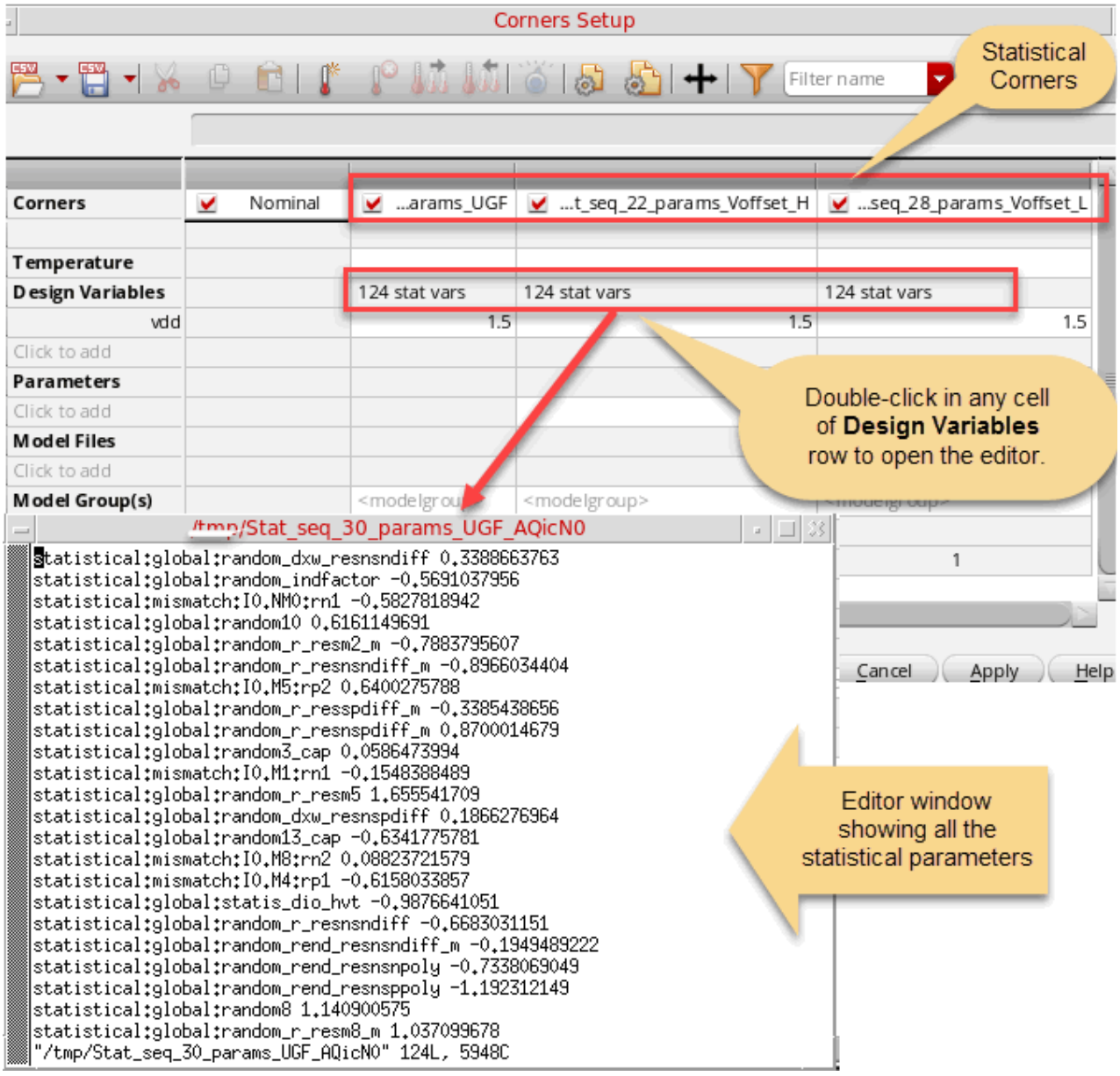
2. Right-click a result for a sample in the *Nominal* column and select *Print Statistical Parameters*.

   The statistical parameters for that sample are displayed in a Results Display Window.


© 2023 Cadence Design Systems, Inc.

## Displaying Statistical Parameters of Statistical Corners

After you create the statistical corners, you can view them in the *Corners* section of the Setup assistant. You can also view them in the Corner Setup form, as shown in the figure below:




The *Design Variables* row displays the statistical parameters count in the `x stat vars` format, where `x` indicates the number of statistical parameters. To view the complete list of these parameters, double-click in the *Design Variable* cell for a statistical corner. An editor window appears that displays the complete list of statistical parameters, as shown in the figure above.


© 2023 Cadence Design Systems, Inc.

© 2023 Cadence Design Systems, Inc.

© 2023 Cadence Design Systems, Inc.

# 9

# Plotting and Printing

This chapter shows how to print and plot simulation data.


- Using the Plot Output Commands on page 767
- Using the Direct Plot Commands on page 770
- Printing Results on page 811
- Printing Components on page 817
- Using SKILL to Display Tabular Data on page 842
- Annotating Simulation Results on page 843


After the simulation run is complete and results are displayed on the Value column of the Outputs Setup tab (for single-point runs) or on the Result tab (for multi-point runs), you can use the Virtuoso Visualization and Analysis XL tool to plot and analyze these results. If the Plot check box in the Outputs Setup is selected, the results are automatically plotted. If only the Save check box is selected, but the result is a waveform, you can manually plot the output waveform. You can use one of the following ways to select and plot simulation results in the Virtuoso Visualization and Analysis XL graph window:

- In the Virtuoso Visualization and Analysis XL Calculator, use the plot icon to plot waveforms.
- In the Results Browser, click right on a node that contains waveforms.
- In the ADE Explorer window, use the *Outputs – To Be Plotted – Select On Design* command to select nets and terminals in the schematic. Use commands in the *Results – Plot Outputs* menu to display the graphs for the selected analysis type.
- In the ADE Explorer window or in the schematic, use the *Results – Direct Plot* command to select nets and terminals in the schematic and to plot a function immediately.

**Note:** When you click a terminal, it gets selected first and then the wire gets selected. Therefore, you can now alternate between the two.


© 2023 Cadence Design Systems, Inc.

If you set up the *Outputs* Setup tab in ADE Explorer, with nets to be plotted, and click the *Plot Outputs* icon during an analysis run, the graph window will pop up and plot the outputs.

Therefore, you get a snapshot of the simulation run upto that time point. You can also use the *Calculator* or the *Results Browser* to plot outputs.

**Note:** After loading a maestro cellview if you want to plot the outputs saved in the setup, you need to run a simulation to generate the results. Otherwise, an error message is displayed. Alternatively, you can open a history from the *Results – Select* menu and plot the outputs.

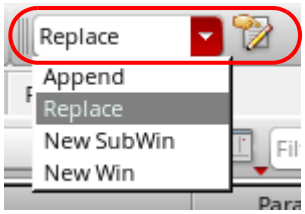

© 2023 Cadence Design Systems, Inc.

## Setting Plotting Mode

The ADE Explorer provides different plotting modes. You can choose a plotting mode depending on your requirements to save and compare plots of different simulation results.

The Plotting toolbar provides the following two options to choose how to plot the results:

- Plotting Mode
- ADE Explorer Plotting/Printing Options Form




### Plotting Mode

You can choose the following plotting modes:

- Append: Appends the new graph to the existing graph.
- Replace: Replaces the existing graph with the new graph. This is the default option.
- New SubWin: Plots the new graph in a new sub window.
- New Win: Plots the new graph in a new window.

You can select the plotting modes for automatic plotting from the *Direct Plot,* the *Virtuoso Visualization and Analysis XL Calculator* and the *Results Browser* windows also.

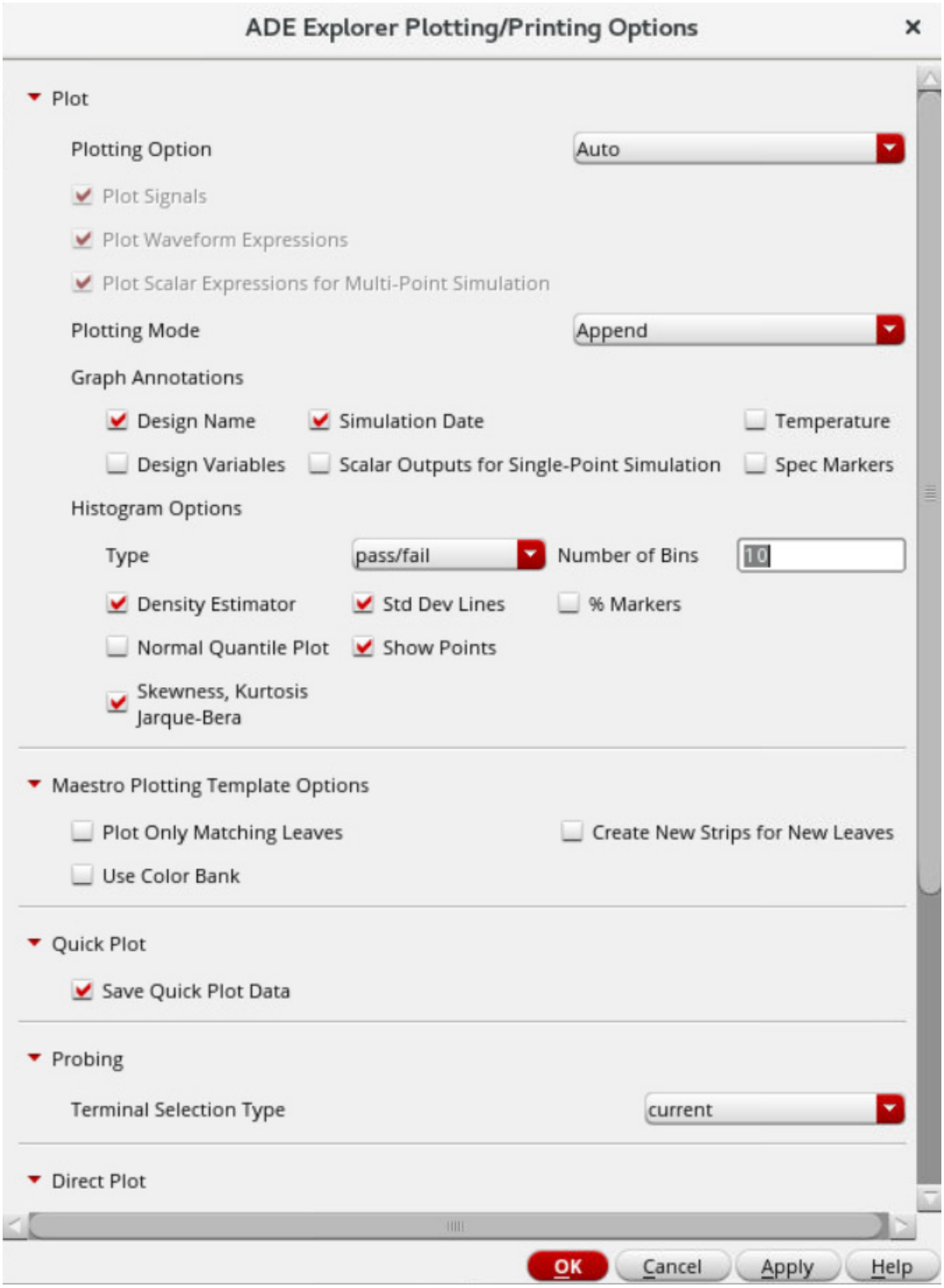
### ADE Explorer Plotting/Printing Options Form

To set the ADE Explorer printing and plotting options, do one of the following:

- Choose *Results – Plotting/Printing Options*.
- Select the plotting button ( ) on the Plotting toolbar.


© 2023 Cadence Design Systems, Inc.

The ADE Explorer Plotting/Printing Options form appears.




You can use this form to set the printing and plotting options.


© 2023 Cadence Design Systems, Inc.

See the following topics for assistance:

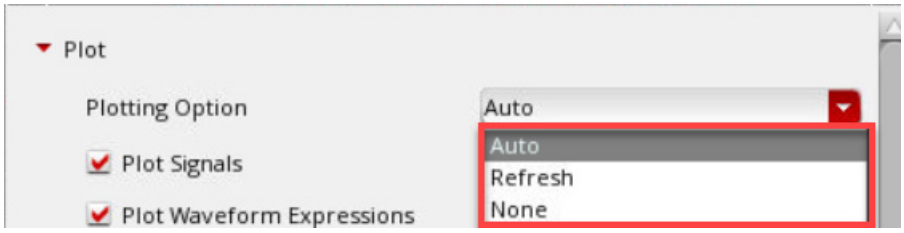
- Specifying the Default Plotting Option
- Specifying the Outputs that Need to be Plotted
- Specifying the Default Plotting Mode
- Specifying Annotations for the Graph Window
- Specifying the Default Direct Plot Options
- Specifying When to Print Results

### *Specifying the Default Plotting Option*

You can specify the default plotting option to be used for the automatic plotting of results after a simulation run is complete and also used to specify which plotting option is used when you click the *Plot all waveforms* button on the Results tab.

To specify the default plotting option,

➡ In the ADE Explorer Plotting/Printing 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. |


© 2023 Cadence Design Systems, Inc.

| Select | To |
|---|---|
| *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.<br><br>When this option is selected, any customization done in the Virtuoso Visualization and Analysis XL 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. |
| *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.<br><br>For more details on how the graphs are refreshed, refer to Refreshing Graphs.<br><br>This mode ignores the following options:<br>■ Plotting mode selected in the *Plotting Mode* drop-down list<br>■ *Plot Signals*, *Plot Waveform Expressions,* and *Plot Scalar Expressions* options<br>■ *Graph annotation* options |

You can also configure the default plotting option by using the `plotType` environment variable.

### *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. |


© 2023 Cadence Design Systems, Inc.

| Select | To |
|---|---|
| *Plot Scalar Expressions for Multi-Point Simulation* | To plot scalar values for a multi-point simulation. |

### *Specifying the Default Plotting Mode*

To specify the default plotting mode, 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 Plotting Mode.

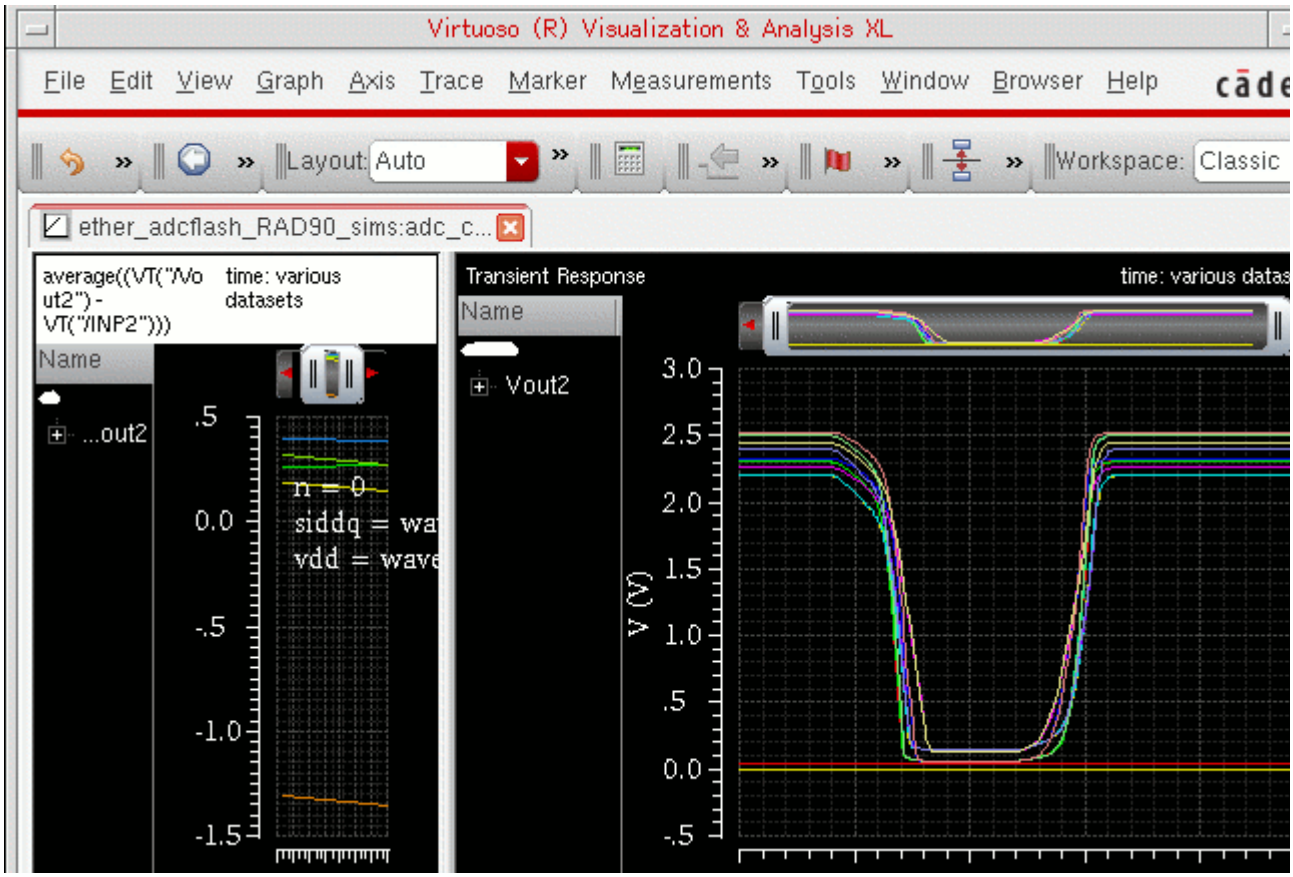
### *Specifying Annotations for the Graph Window*

To specify annotations for the graph window, do the following:

1. Select one or more of the following *Annotations* check boxes:
   - *Design Name* — Annotates design name in the title banner of the waveform window.
   - *Simulation Date* — Annotates date and time of simulation in the title banner of the waveform window.
   - *Temperature* — Annotates simulation temperature in the plot area of the waveform window.
   - *Design Variables* — Annotates design variables and their values in the plot area of the waveform window.
   - *Scalar Outputs for Single-Point Simulation* — Annotates scalar values for a single-point simulation.
   - Spec Markers — Annotates spec markers in the plot area of the waveform window.
2. Click *Apply*.

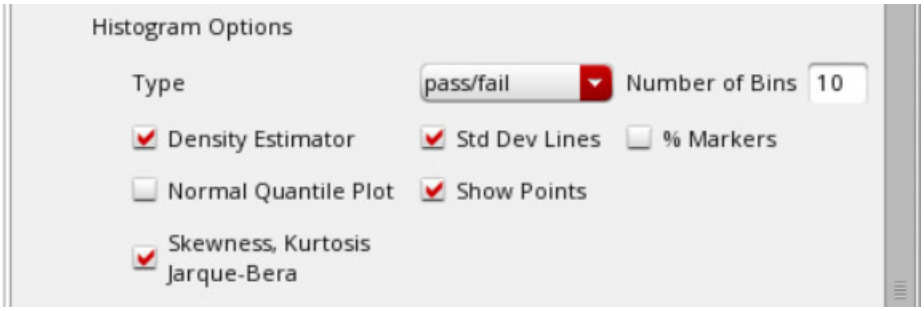

© 2023 Cadence Design Systems, Inc.

Here is a waveform window with all annotations applied:




### *Specifying Histogram Options*

To specify the histogram options, do the following in the *Histogram Options* section:



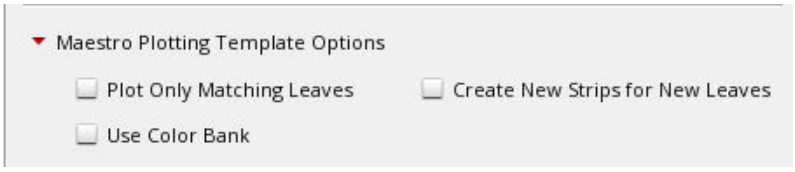

- In the *Type* drop-down, specify the type of histogram to be plotted. The available types are `standard`, `pass/fail`, `cumulative line`, and `cumulative box`. The default type is `pass/fail`.


© 2023 Cadence Design Systems, Inc.

- In the *Number of Bins* field, specify the bins. Default value is `10`.
- Select the following checkboxes:
  - *Density Estimator*
  - *Std Dev Lines*
  - *% Markers*
  - *Normal Qualtile Plo*t
  - *Show Points*

### *Specifying Maestro Plotting Template Options*

To enable or disable default maestro plotting template options, do the following in the *Maestro Plotting Template Options* group box:




| Select | To |
|---|---|
| *Plot Only Matching Leaves* | Plot matching leaves with plotting templates. |
| *Create New Strips for New Leaves* | Plot new leaves for signals in new strips. |
| *Use Color Bank* | Plot signals using the default color bank. |

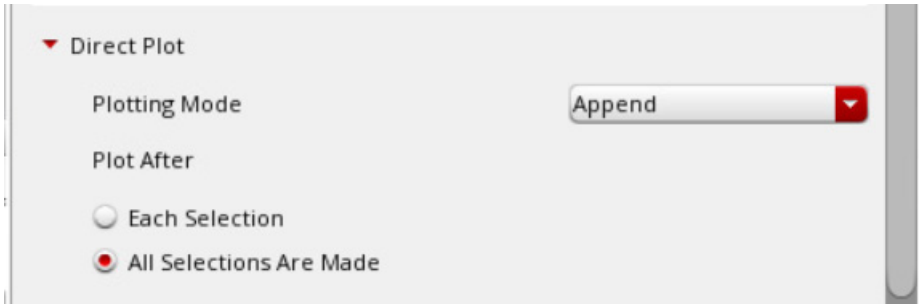
### *Specifying Whether to Save Quick Plot Data*

To save the data for a quick plot, select the *Save Quick Plot* data check box in the *Quick Plot* section. This field is enabled by default. When this field is not selected, the *Quick Plot* option does not work for the output signals or expressions.


© 2023 Cadence Design Systems, Inc.

### *Specifying the Default Direct Plot Options*

To specify the default direct plot mode options, do the following in the Direct Plot drop-down:



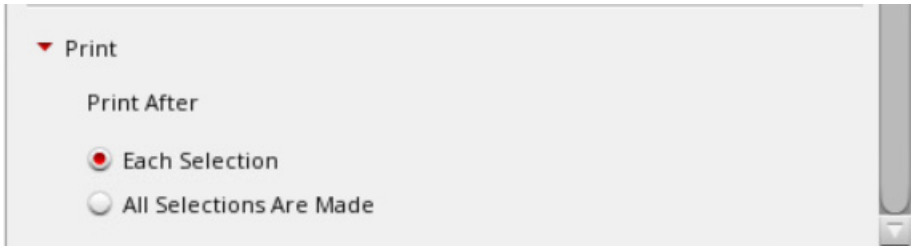

1. Select the plot mode from the *Plotting Mode* drop-down list.

   For more information about plot modes, see Plotting Mode.

2. 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.
3. Click *Apply*.

### *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*.


© 2023 Cadence Design Systems, Inc.

### *Enabling the Distributed Plot Service*

To enable the distributed plot service:

1. Select the *Enable Distributed Plot* check box.
2. (Optional) In the *Linger Time (seconds)* field, specify the time in seconds for which the distributed plot process waits before exiting when there is no active plotting window is visible. The default value is 60 seconds. It is recommended not to set the linger time to less than 30 seconds.
3. Click *Apply*.

## Distributed Plot Commands

The following table describes the distributed plot commands that are displayed when you click the arrow next to the icon in the Run toolbar of ADE Explorer.

| Command | Description |
|---|---|
| *Open Graph* | Opens the Distributed Plot graph window. |
| *Job Distribution* | Opens the Job Policy Selection for Distributed Plot form. This form lets you select the job policy that you have set up earlier in the Job Policy form. |
| *Show Process List* | Displays information about the currently running processes in an xterm window. |
| *Show Assistant* | Opens the Distributed Plot assistant. The Distributed Plot assistant lets you view and manage various plotting jobs |
| *Exit* | Exits the distributed command menu. |

These commands become unavailable if you disable distributed plot. The icon indicates that distributed plot is disabled.


© 2023 Cadence Design Systems, Inc.

### *Related Topics*

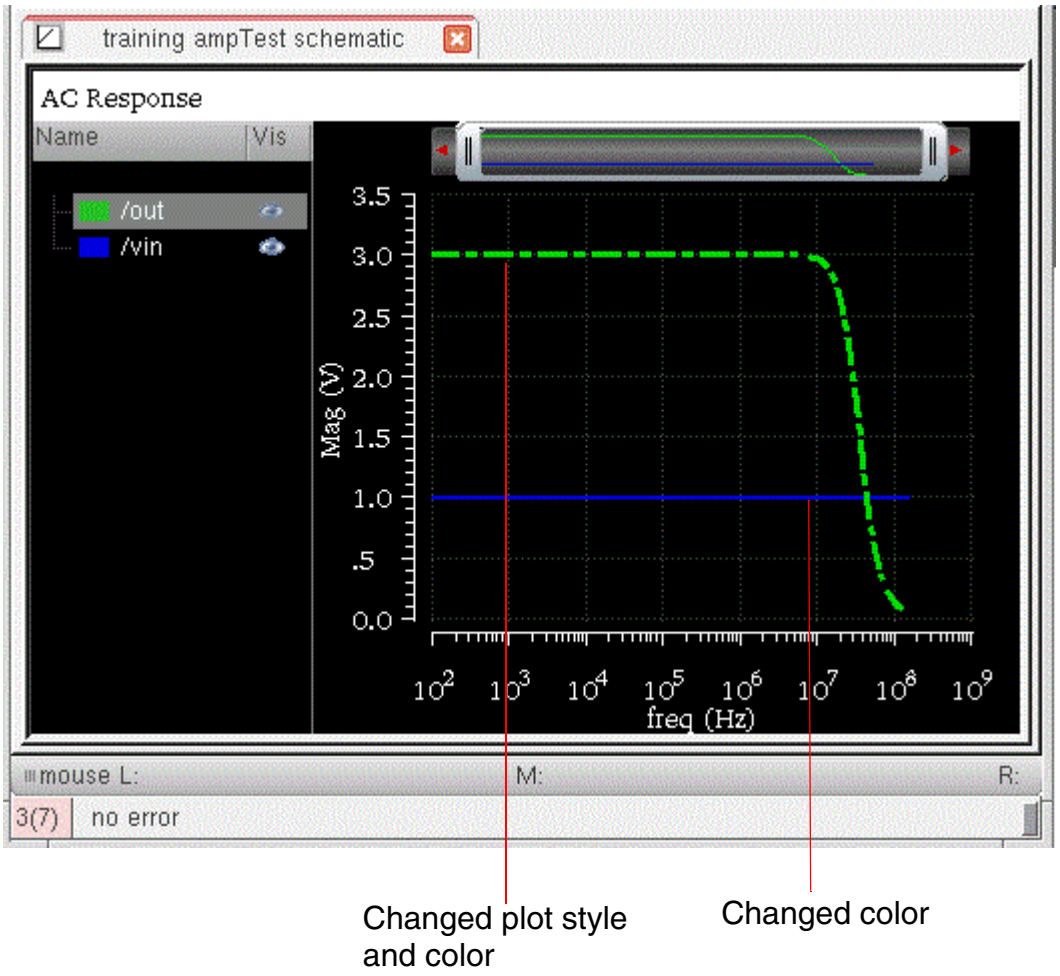
Distributed Plot


© 2023 Cadence Design Systems, Inc.

## Refreshing Graphs

Graphs can be refreshed using the *Refresh* option in the *Plotting Options* drop-down box in the ADE Explorer Plotting/Printing Options form. This option updates the open graphs in Virtuoso Visualization and Analysis XL with the new simulation results, and retains the graph and trace settings.

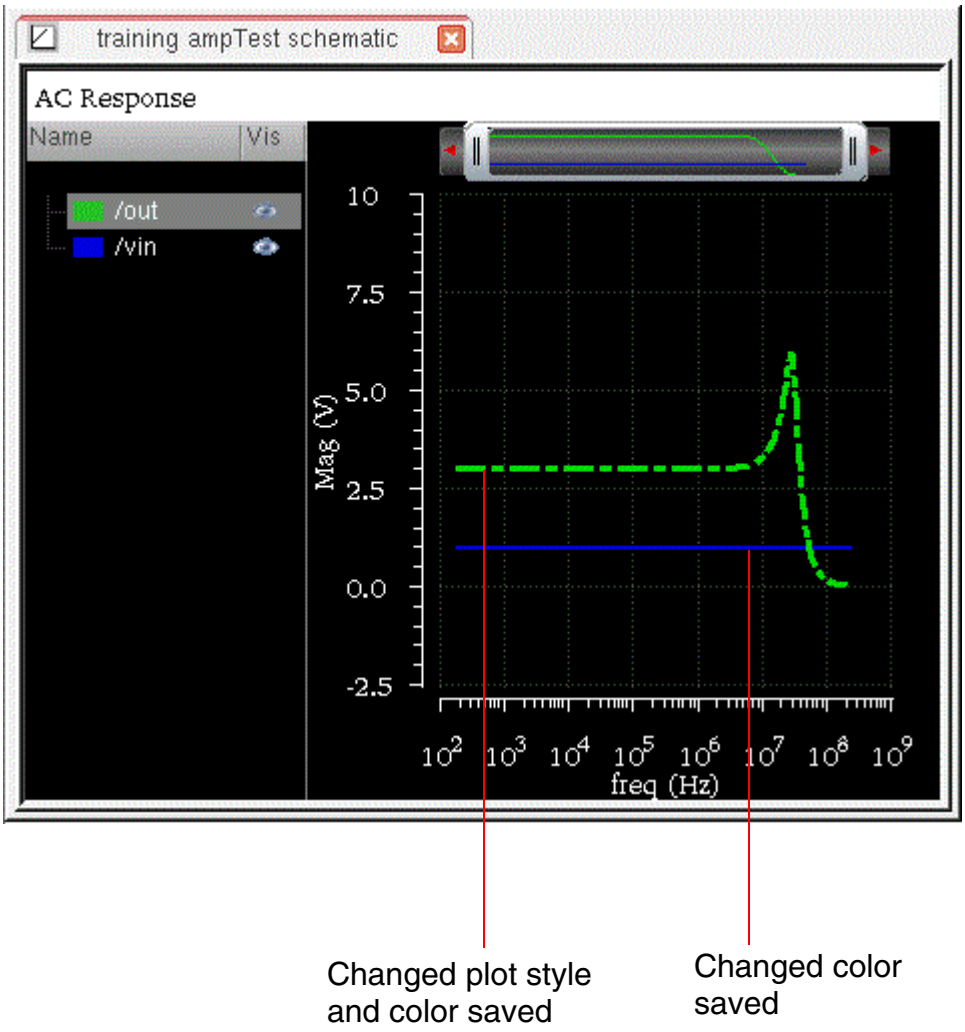
You can use this option to review graphs across different simulation runs. For example, consider you want to run multiple simulations with different values of the design variable CAP. In the first run, plot CAP for `800f`. After the simulation results are plotted in the graph, customize the plots, as shown in the figure below.




In the next run, use CAP=200f to run the simulation. Using the Refresh option, the simulation results are updated in the same graph.


© 2023 Cadence Design Systems, Inc.

**Note:** The settings done to the plots are retained, as shown in the following figure.




**Note:** You can use File – Reload to reload the plotted graphs based on the results from a new in-context results directory selected in the Results Browser. If you rerun the simulation even without changing the corner settings, Reload plots traces from all the corners in the graph. To know more, see Reloading Graph When Opened From Within ADE L, ADE XL, ADE Explorer or ADE Assembler in the Virtuoso Visualization and Analysis XL User Guide.

The following sections describe how the graphs are refreshed using the *Refresh* option in different scenarios:

- Refresh Plotting with Varying Analysis
- Refresh Plotting with Varying Outputs


© 2023 Cadence Design Systems, Inc.

- Refresh Plotting with Parametric Simulation

### Graph Settings Supported by the Refresh Plotting Mode

The Virtuoso Visualization and Analysis XL graph window saves and maintains the following settings for the graphs that are generated for a common set of variable combinations across simulation runs:

- Trace color, type, style, width, or symbols
- Visibility status of graphs
- Axes settings
- Pan and zoom settings
- Graph layout
- Strip layout
- Makers and marker locations

### Graph Settings Not Supported by the Refresh Plotting Mode

Currently, the following graph settings are not supported:

- Swapping of sweep variable on the X-axis
- Addition of a new graph window
- Any signal, expression or measurement that you directly added to the graph and is not specified in the *Outputs Setup* tab in ADE Explorer window.

*Important*

To retain additional or derived graphs and their settings, you can send plot from the graph to the ADE Explorer *Outputs Setup* 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*.

### Refresh Plotting with Varying Analysis

The following table describes how graphs are refreshed when analysis are varied across different simulation runs:


© 2023 Cadence Design Systems, Inc.

**Table 9-1 Effect of Analysis Variations on Graph Settings with Refresh Option**

| Analysis 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.<br>■ Graphs for the analysis that are common across different runs are updated with 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 Outputs

The following table describes how graphs are refreshed when outputs are varied across different simulation runs:

**Table 9-2 Effect of Output Variations on Graph Settings with Refresh Option**

| Analysis Variation | Effect on Graph Settings |
|---|---|
| Add or enable an output | ■ Plot for the new output is added in a new subwindow.<br>■ Other existing graphs are refreshed and their trace settings are maintained. |
| Delete or disable an output | ■ The plot for the deleted or disabled output is removed.<br>■ Other existing graphs are refreshed and their trace settings are maintained. |

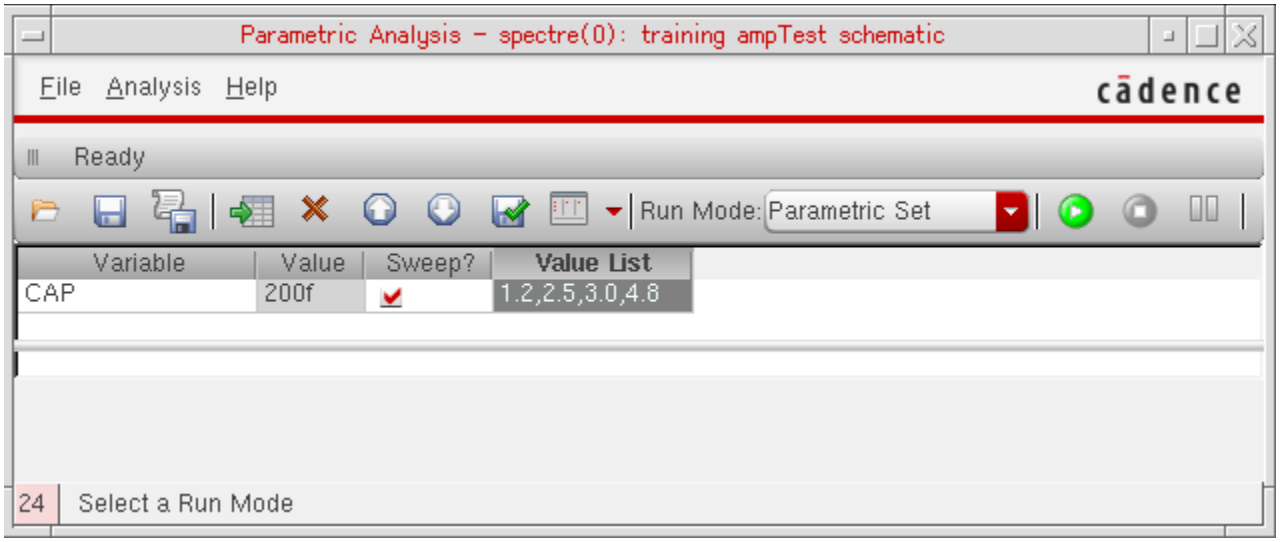
**Note:** The *Refresh* option does not update graphs for MATLAB measurements.

### Refresh Plotting with Parametric Simulation

If you perform parametric simulation runs, the graphs are updated with new simulation results for common sweep values between subsequent runs. Traces corresponding to new sweep values are added to the same graph. Traces corresponding to unmatched sweep values are deleted.


© 2023 Cadence Design Systems, Inc.

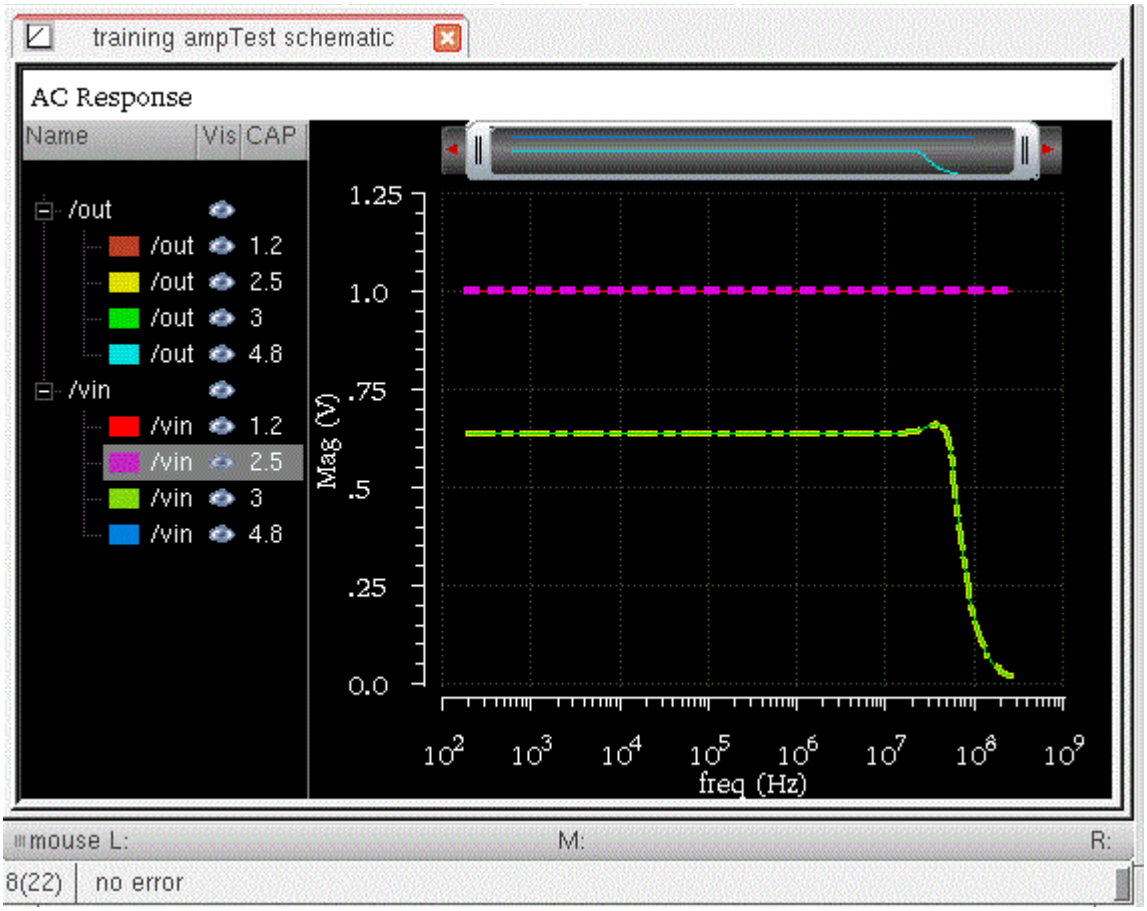
For example, set the variables as shown below.





© 2023 Cadence Design Systems, Inc.

The graph plotted in this case includes plots for all four sweep values for CAP, as shown below.



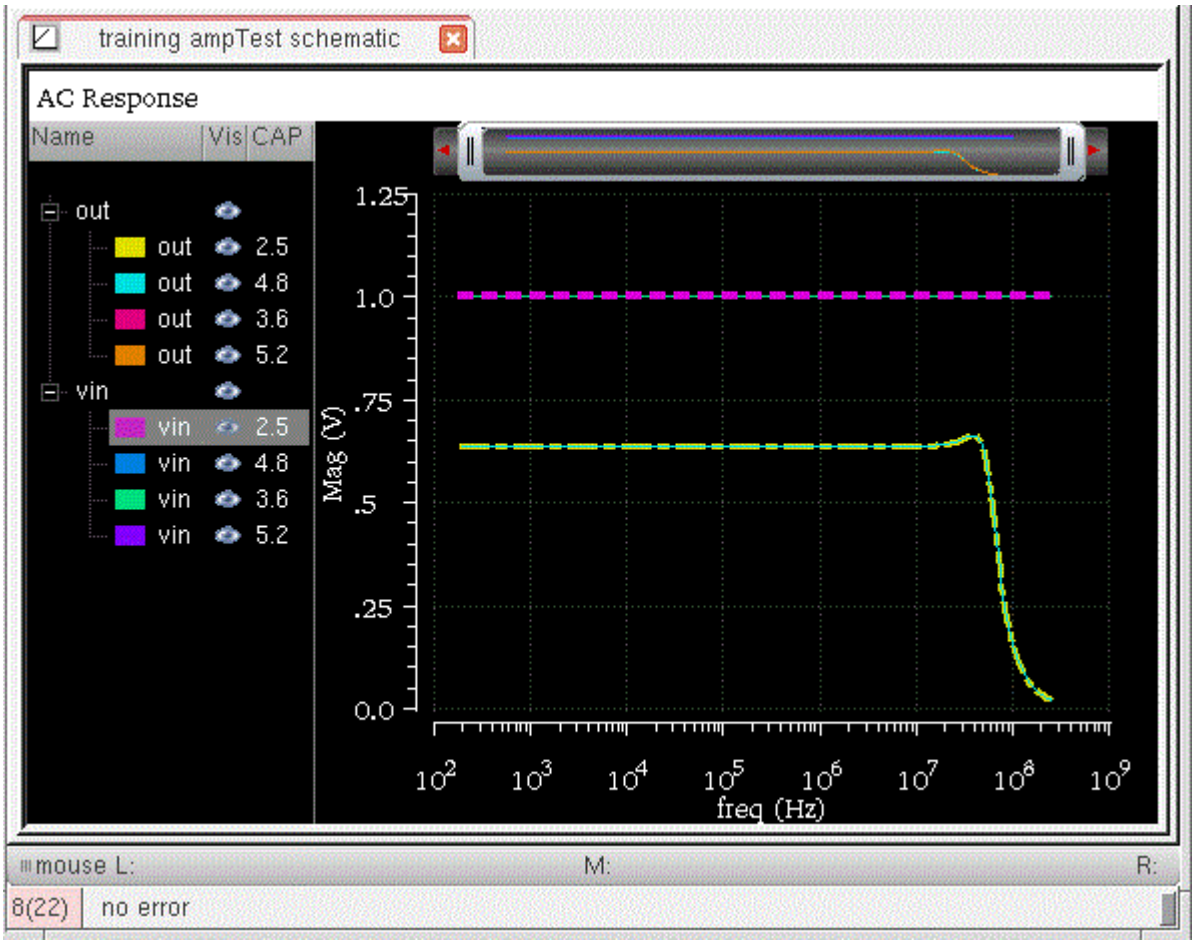

For the next simulation run, change the sweep values for CAP to `2.5:4.8:3.6:5.2`. Note the change in the values of CAP for which the simulation is run (mismatch values are underlined), as shown below.

Values of CAP in the first run: `1.2`, `2.5`, `3.0`, and `4.8`

Values of CAP in the second run: `2.5`, `4.8`, `3.6`, and `5.2`

With the Refresh plotting, the traces corresponding to CAP = `2.5` and `4.8` are updated with new simulation data. Traces corresponding to CAP = `1.2` and `2.0` are removed. Traces corresponding to CAP = `3.6` and `5.2` are added.


© 2023 Cadence Design Systems, Inc.




### *Unsupported Scenarios of Sweep Variations*

In the following scenarios, the graphs for common sweep values for the swept variables are not refreshed:

- If you run parametric simulations after you run normal simulations in ADE Explorer window, graphs are not refreshed.
- Change from single sweep variable to multiple sweep variables. For example, in the first simulation run, you sweep variable `x`, and in the subsequent runs, you sweep variables `x` and `y`.
- Change in the sweep variable. For example, in the first run, you sweep variable $x$ and in the subsequent runs, you sweep variable `y`.


© 2023 Cadence Design Systems, Inc.

## Performing Quick Plotting

When the ADE Explorer setup includes multiple sweep or corner points, or the transient simulations are time-consuming, plotting output waveforms through the *Plot All* option could consume significant time and memory. In this case, you can use the *Quick Plot* option to plot outputs faster and with much lesser memory usage (data size <1 percent of PSF XL). The quick plot is faster because it uses a reduced set of data points as opposed to the *Plot All* option that generates a full view covering all data points. The quick plot compresses data, and therefore, saves memory.

However, quick plots have lower resolution as compared to the full view that is generated using the Plot All option. When you send the quick plot waveforms to Calculator, the full plot data is loaded so that the calculations are not affected. In addition, if you perform any operations on quick plot signals using Results Browser, such as `Diff`, the output waveform is also of low resolution.

**Note:** The quick plot option is available only when the simulation run is complete and the output results are generated. Currently, this option works only for the transient signals and waveform expressions, and is available only for the `Detail` and `Detail-Transpose` results views. In addition, quick plot can be generated for all signals and expressions except for histograms. The *Quick Plot* option is disabled when you re-evaluate the results for expression after the simulation run is complete.

While running a Monte Carlo simulation, quick plotting is not supported if the *Save Waveforms (Simulation Data)* check box is not selected on the Monte Carlo options form. In this case, an error message is displayed in CIW.

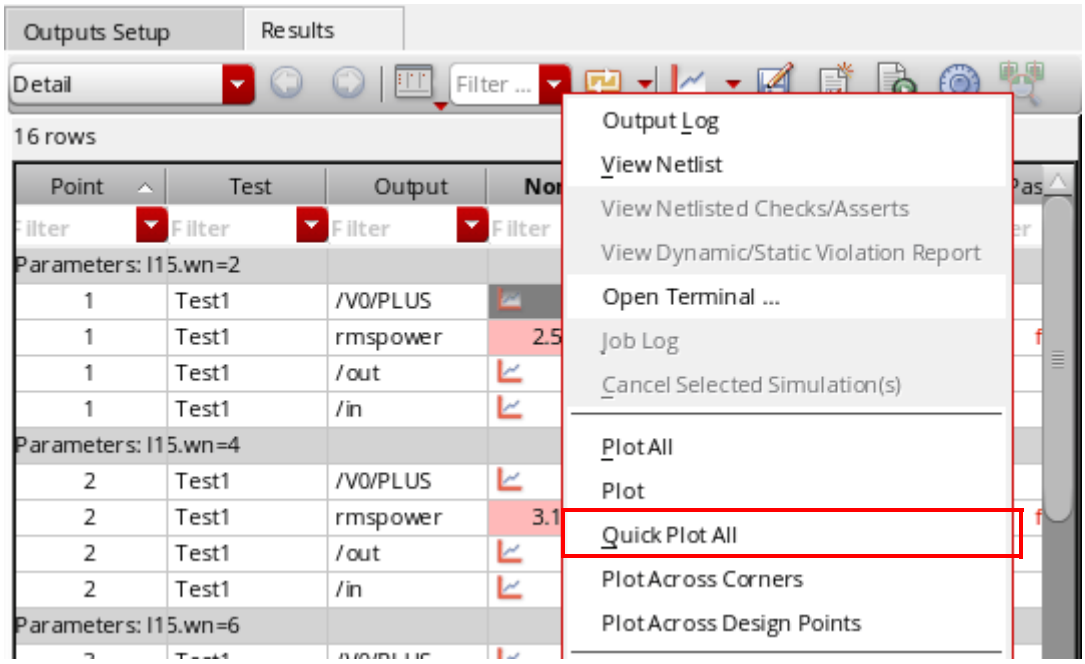
**Note:** If you perform the quick plotting of scalar signals, the full view plot is generated.

To generate the quick plot for a selected signal or expression:

- Right-click the waveform icon for a signal or waveform expression output and choose *Quick Plot All*.

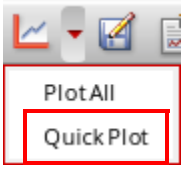

© 2023 Cadence Design Systems, Inc.

A quick plot is generated in the Virtuoso Visualization and Analysis XL window for the selected output signal.




To generate the quick plot for all the signals and expressions:

➡ In the *Plot All* drop-down list on the *Results* tab, click *Quick Plot*.




The quick plot waveforms for all the signal outputs are generated in the graph window.

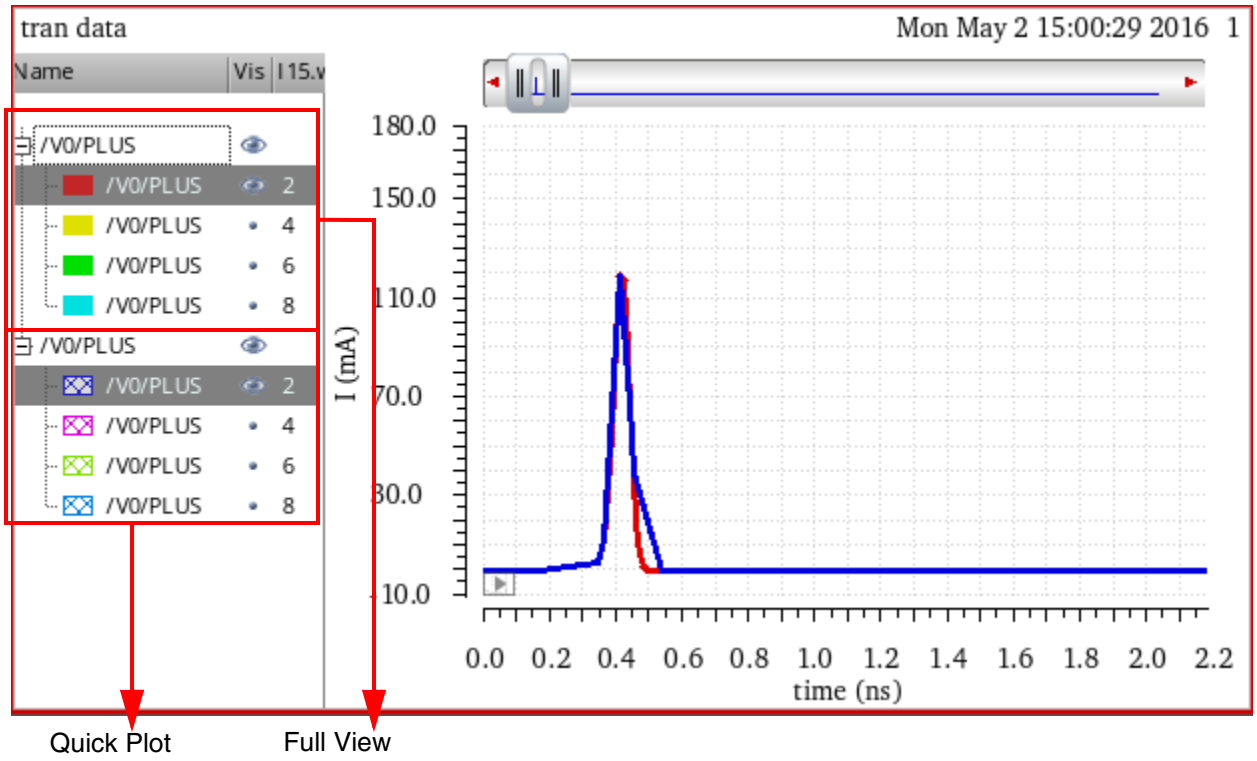
*Important*

The quick plots are generated only when the *Save Quick Plot Data* option is selected on the ADE Explorer Plotting/Printing Options form, which appears when you choose *Results – Plotting/Printing Options*. If this option is disabled, an error message is displayed in CIW. For more information, see Specifying Whether to Save Quick Plot Data on page 753.

The figure below shows the waveforms from normal plot (full view) displayed in red and quick plot displayed in blue for a current signal, `/V0/PLUS`. The quick plot waveform includes fewer points as compared to the full view. When the full view and quick plot are displayed in the


© 2023 Cadence Design Systems, Inc.

same window in append mode, you can zoom into the plots to compare them and see the differences between them. In addition, notice that the legend of the quick plot is different from the full view, which helps differentiating between the two plots.




The equivalent full data is loaded when you send the quick plot waveform data to Calculator. The loaded waveform is a leaf, for example:

```
leafValue( IT("/V0/PLUS") "I15.wn" 2 )
```

In the Results Browser, the signals are displayed in the order they are computed and may not match the order in which they are viewed in the full view data.

The saved quick plot data can be loaded into graph and is located at:

```
<Interactive>/wavedb/<test>.vqv
```


© 2023 Cadence Design Systems, Inc.

You can also generate a full view from the quick plot data. To do this, right-click the quick plot waveform and choose *Quick Plot Expand to – New Window/New Subwindow/Current Window*.

**Note:** You can also read the *Virtuosity: Can I Speed up My Plots?* blog for more information about quick plotting.

# Using the Plot Output Commands

To plot the outputs, choose one of the below listed commands in the *Results – Plot Outputs* menu.

| | |
|---|---|
| *Transient* | Plots the transient response for each node |
| *AC* | Plots the AC response for each node |
| *DC* | Plots the DC sweep response for each node |
| *Noise* | Plots the squared noise voltage for each node |
| *Expressions* | Plots the waveforms for expressions you define in the Setting Outputs form. For more information, see Saving Output Expression on page 408. |

## Plotting the Current or Restored Results

To plot the most recent results in the graph window:

1. Choose *Outputs – To Be Plotted – Select On Design*.

   Nodes and terminals that you have already selected are now highlighted.

2. In the schematic window, select one or more nodes or terminals.
3. Press the `Escape` key when you finish selecting nodes and terminals.

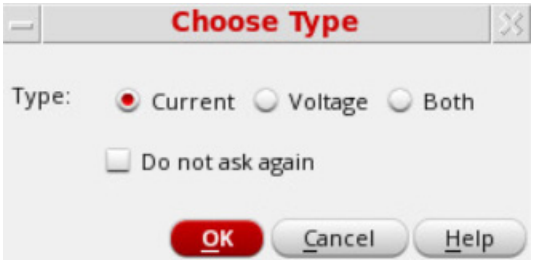

© 2023 Cadence Design Systems, Inc.

4. You can also add a signal or expression manually to the outputs setup.

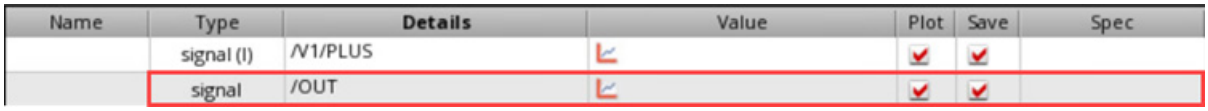
   The terminal type for the added signal or expression is determined by the *Terminal Selection Type* set in the *Plotting/Printing Options* form.

   For example, manually add an instance terminal `/I0/M3/B` to the outputs setup. Depending on the specified *Terminal Selection Type*, the following four outcomes are possible:

   - *Current*: Implies that a `signal (I)` is added.
   - *Voltage:* Implies that a `signal (V)` is added.
   - *Both:* Implies that a `signal (I)` and `signal (V)` are added.
   - *Prompt:* In this case, the following form is displayed, where you can select the type of signal to save.

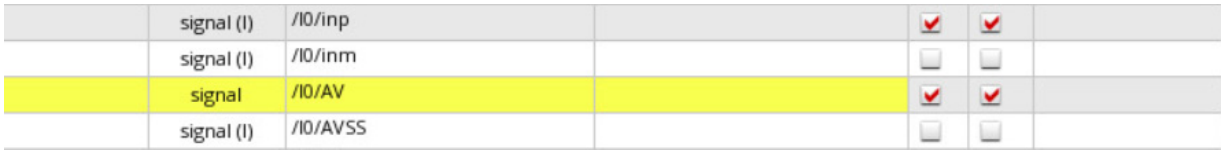



**Note:** On selecting a net, the output *Type* is set to `signal` as shown in the figure below.



| Name | Type | Details | Value | Plot | Save | Spec |
|---|---|---|---|---|---|---|
| | signal (I) | /V1/PLUS | | ✓ | ✓ | |
| | signal | /OUT | | ✓ | ✓ | |

When you add new expressions, the *Plot* check box is selected. On adding new signals, both the *Plot* and *Save* check boxes are selected. If an added signal is not found in the schematic design hierarchy then it is considered invalid and gets highlighted in yellow in the outputs setup.



| | | | | | | |
|---|---|---|---|---|---|---|
| | signal (I) | /I0/inp | | ✓ | ✓ | |
| | signal (I) | /I0/inm | | ☐ | ☐ | |
| | signal | /I0/AV | | ✓ | ✓ | |
| | signal (I) | /I0/AVSS | | ☐ | ☐ | |


© 2023 Cadence Design Systems, Inc.

If an expression is created on a non-existent net or terminal or if the expression contains signals that do not exist in the schematic, the expression is highlighted in yellow in the outputs setup.

| | | | | |
|---|---|---|---|---|
| AC | Signal1 | signal | /IDontExist | point |
| AC | Expression1 | expr | VT("/NotInDesign") | point |
| AC | Expression2 | expr | (VT("/IDontExist") - VT("/NotInDesign")) | point |
| AC | | expr | (Expression1 + Expression2) | point |
| AC | | expr | (Expression1 + VT("/OUT")) | point |
| AC | | expr | VT("/OUT") | point |

5. Choose a *Results – Plot Outputs* command in the ADE Explorer window.

   The system plots the results you selected in the current graph window or opens a new graph window if one is not open.

To plot all of the available results at once,

➤ Click the *Plot Outputs* icon on the Run toolbar.

**Note:** You can plot only saved voltages and currents.

When you choose *Outputs – To be Plotted – Select on Design*, and then select an iterated instance in the schematic, the *Select instTerm IN on iterated inst* form is displayed.

If you select a bus signal, the *Select bit from bus* form is displayed.

These forms enable you to select from one to all bits of an iterated item. When you select the top element in the list box, all the individual bits are selected. You can also select an individual bit with the left mouse button.

**Note:** `Ctrl-Left` mouse will toggle selection of an item. `Shift-Left` mouse will select all items between the last selected item and the current item.

## Plotting Parasitic Simulation Results

When you plot the results of a parasitic simulation, only terminals and device pins can be mapped from the schematic to the extracted view.

To select results in the schematic while parasitic simulation is enabled,

1. In the ADE Explorer window, choose *Outputs – To Be Plotted – Select on Design*.
2. In the schematic, select a terminal or pin, or a wire near a terminal or pin.

   If you select a point in the middle of a wire, the system chooses the nearest terminal or device pin and you might not get the right data.


© 2023 Cadence Design Systems, Inc.

The system draws an X to mark the point you selected.

3. Choose a *Results – Plot Output* command.

   The color of the waveform matches the color of the X.

You cannot probe nets that connect only sources and loads because these nets do not exist on the extracted view. You also cannot probe nets between parasitic components that were removed by selective annotation because these nets were removed when the selected view was built.

# Using the Direct Plot Commands

You can plot common waveforms quickly in the ADE Explorer window using the *Direct Plot* commands. With these commands, you do not need to use the calculator to create common expressions and you do not need to add the nets or terminals to the plot set.

To use Direct Plot,

➤ Choose *Results – Direct Plot – Main Form*. This brings up the unified *Direct Plot Form* that changes dynamically depending on the analysis that was performed.

or,

➤ Choose the *Results – Direct Plot* commands. The commands are as follows:

| This option… | Plots this curve… |
|---|---|
| *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<br>10log(V1) |
| *AC db20* | The magnitude of selected signals on a decibel scale<br>20log(V1) |


© 2023 Cadence Design Systems, Inc.

| This option… | Plots this curve… |
|---|---|
| *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<br>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 |
| *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.<br>**Note:** Noise figure is calculated by Spectre if a port is selected as the input source for noise analysis. If a port is not selected as the input, noise figure data is not available.<br>The mathematical noise-figure expression is:<br>$NF = 10\log\left(\frac{VN2*\lvert Vin\rvert}{C*\lvert Vout\rvert*\mathrm{R}}\right)$<br>$VN2$ = The noise voltage<br>$Vin$ = The voltage at the input node<br>$Vout$ = The voltage at the output node<br>$R$ = The source resistor value<br>$C$ = 1.61e-20 $\cong 4k\mathrm{T}\Delta\mathrm{f}$ with $T$ = 291.5K *and* $\Delta\mathrm{f}$ = 1 |
| *DC* | DC sweep voltage or current waveform |


© 2023 Cadence Design Systems, Inc.

To use *Direct Plot* commands,

1. Choose a command from the *Results – Direct Plot* menu.

   If necessary for the command, a form appears.

   The graph window opens. If a graph window was already open, it becomes active.

2. Look in the schematic window for a prompt.

   The prompt tells you what to select in the schematic.

3. Select the signals necessary for the function and press the `Escape` key.

   The system plots the signals. The system shuffles windows automatically, so that the graph window is in front.

There are two modes for the *Direct Plot* commands:

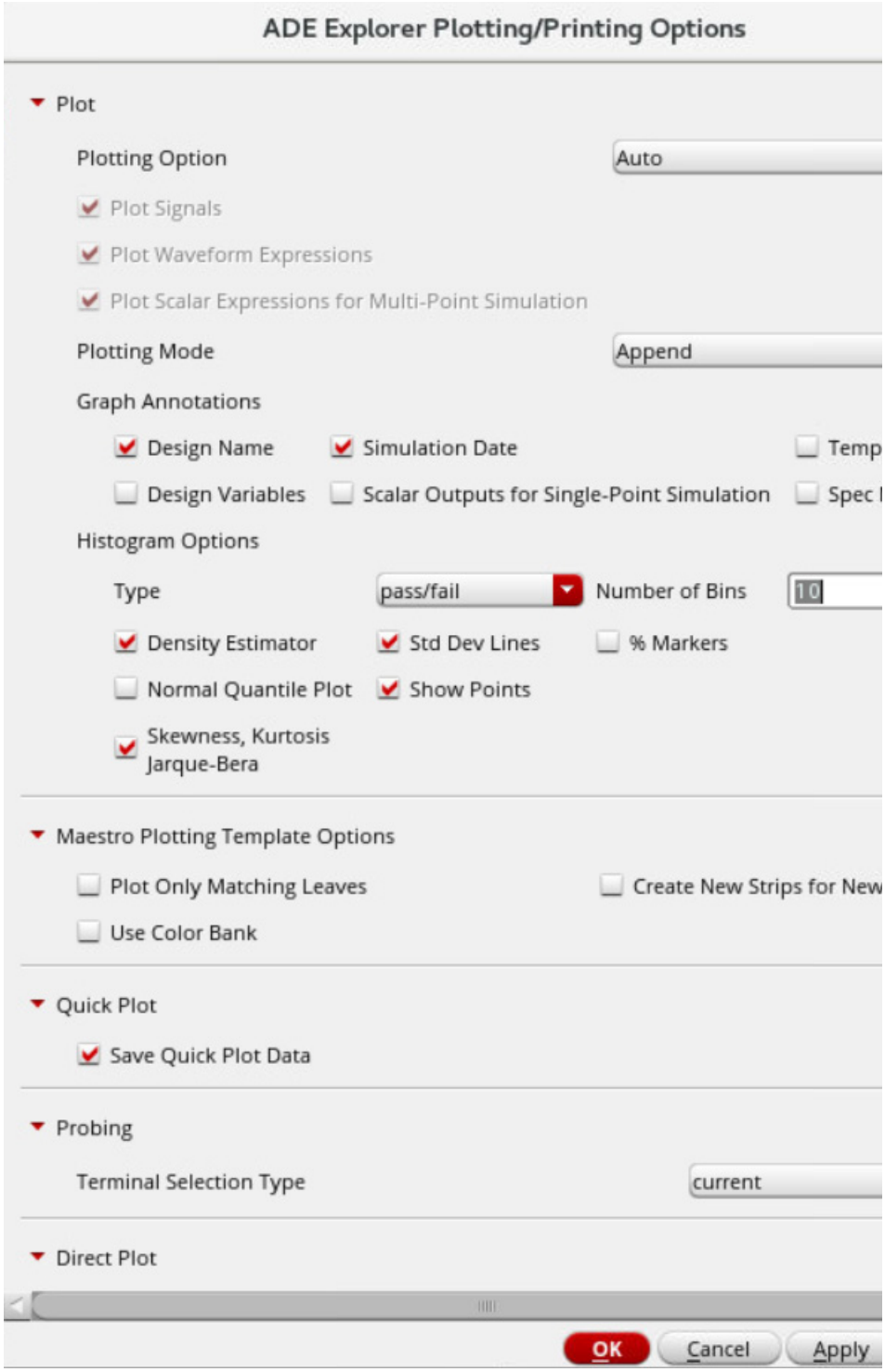
- Plotting one signal at a time immediately after you select the signal
- Plotting several signals together after you press the `Escape` key

To choose the mode,

1. Choose *Results – Plotting/PrintingPlotting/Printing Options*.


© 2023 Cadence Design Systems, Inc.

The ADE Explorer Plotting/Printing Options form appears. For more information about the form, see ADE Explorer Plotting/Printing Options Form on page 747.





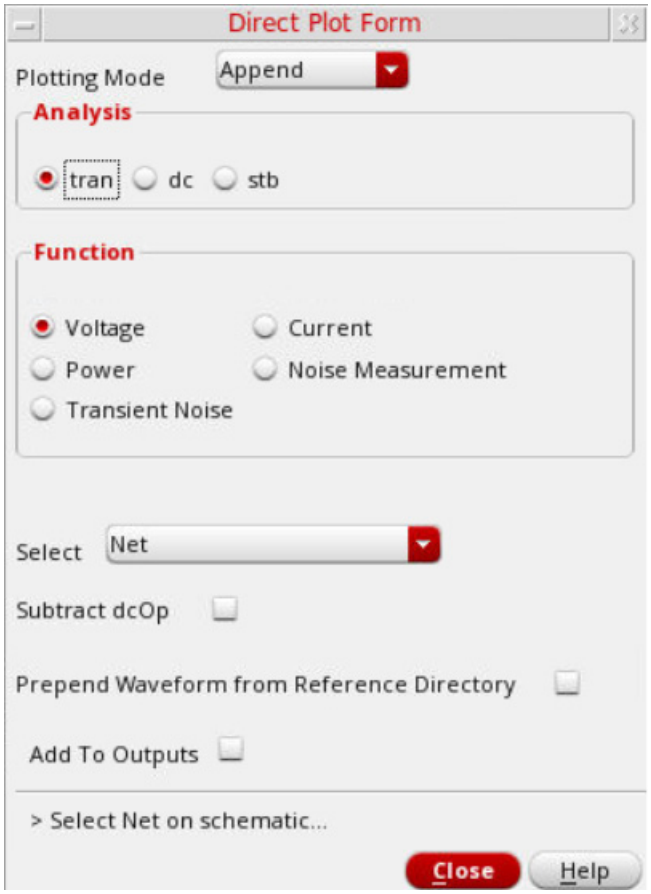
© 2023 Cadence Design Systems, Inc.

2. Choose the required options from the *Direct Plot* group box and click *OK*.


© 2023 Cadence Design Systems, Inc.

## Using the Direct Plot Main Form

When you choose *Results – Direct Plot – Main Form*, the following form appears:

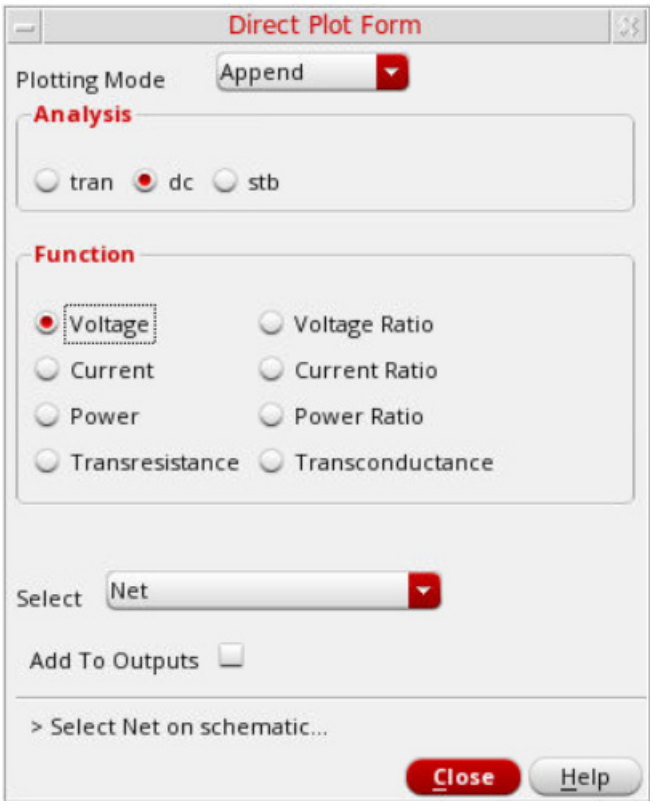



## For DC

1. Choose *Results – Direct Plot – Main Form*. The Direct Plot Form appears.
2. Select *dc* in the *Analysis* section.


© 2023 Cadence Design Systems, Inc.

The Direct Plot Form is updated, as shown below:




3. Specify the *Plotting Mode*. You can specify:

   - *Append* to append the new signal to the current graph.

     **Note:** The *Append* mode is not recommended for plots with different scales and units for the X axis. It can give strange results because *Virtuoso Visualization and Analysis XL* opens a new subwindow to plot any data that does not match the units and range of the existing traces.

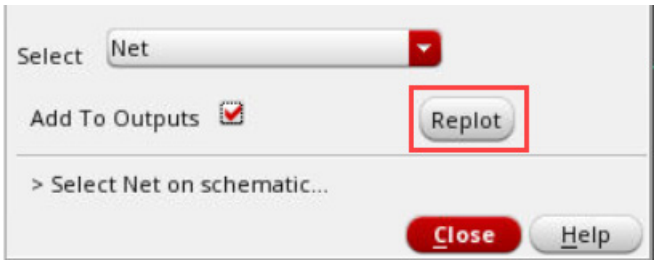
   - *Replace* to replace the current graph with the new signal. This is the default option.
   - *New SubWin* to plot the signal in a new subwindow.
   - *New Win* to plot the signal in a new window.

4. The functions that are available are: *Voltage*, *Voltage Ratio*, *Current*, *Current Ratio*, *Power, Power Ratio, Transresistance,* and *Transconductance*. Based on the selected function and available data, the form changes dynamically to display the applicable options.


© 2023 Cadence Design Systems, Inc.

Enable *Add To Output*s to add expressions for the results to the outputs section and plot in the mode that you selected.

**Note:** On deselecting this check box, it turns into a button that can then be clicked to add the expressions for the results to the outputs section. If this check box is reselected, on clicking *Replot*, the results are plotted as well as added to the outputs section.




Based on the selected function and available data, the form changes dynamically to display the applicable options.

- For *Voltage,* choose the nets and terminals to plot. You can select one of the following options to display the graphs for the selected analysis type.
  - *Net*: Lets you select nets from the schematic.
  - *Terminal*: Lets you select instance terminals from the schematic
  - *Differential Nets*: Lets you select differential nets from the schematic.
  - *Instance with 2 Terminals:* Lets you select an instance with two terminals in the schematic.
- For *Current,* choose the terminals to plot. You can select one of the following options to display the graphs for the selected analysis type.
  - *Terminal*: Lets you select instance terminals from the schematic.
  - *Differential Terminals*: Lets you select differential terminals from the schematic.
- For *Power*, choose the nets and terminals to plot. You can select one of the following options to display the graphs for the selected analysis type.
  - *Terminal*: Lets you select instance terminals from the schematic.
  - *Terminal and V-Reference Terminal*: Lets you select positive and negative instance terminals from the schematic.
  - *Instance with 2 Terminals*: Lets you select an instance with two terminals from the schematic.


© 2023 Cadence Design Systems, Inc.

- *Total Power*: Plots the total power of the circuit.
- Instance: Lets you select instances from the schematic.

- For *Transresistance*, choose the nets and terminals from the schematic. This function gives you the transresistance of the selected net voltage over the selected device node current. From the *Select* drop-down list, select one of the following options.
  - *Numerator Net and Denominator Terminal*: Lets you select the numerator net and the denominator instance terminal from the schematic.
  - *+-Numerator Net and Denominator Terminal*: Lets you select the differential numerator net and the denominator instance terminal from the schematic.
- For *Voltage Ratio*, choose the nets and terminals from the schematic. This function gives you the voltage ratio of two net voltages, or a pair of differential net voltages. From the *Select* drop-down list, select one of the following options.
  - *Numerator Net and Denominator Nets*: Lets you select the numerator and the denominator net from the schematic.
  - *+-Numerator and +- Denominator Nets*: Lets you select pairs of differential numerator nets and denominator nets from the schematic.
- For *Current Ratio*, choose the terminals from the schematic. This function gives you the current ratio of two instance terminal currents.
- For *Power Ratio*, choose the nets and terminals from the schematic. From the *Select* drop-down list, select one of the following options.
  - *Numerator Net and Denominator Terminals*: Lets you select the numerator and the denominator instance terminals from the schematic.
  - *+-Numerator and +- Denominator Terminals*: Lets you select pairs of differential numerator instance terminals and denominator instance terminals from the schematic.
  - *Num. and Denom. Instances with 2 Terminals*: Lets you select the numerator and denominator instances with two terminals from the schematic.
- For *Transconductance*, choose the nets and terminals from the schematic. This function gives you the transconductance of the selected device node current over the selected device net voltage. From the *Select* drop-down list, select one of the following options.

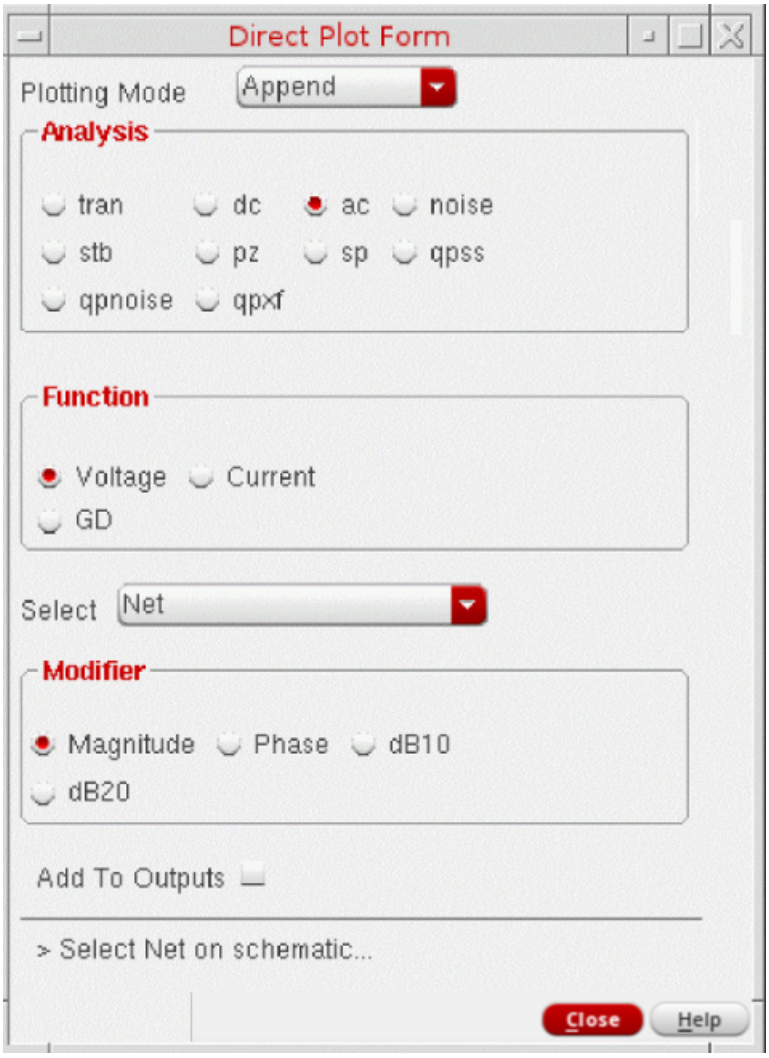

© 2023 Cadence Design Systems, Inc.

- *Numerator Terminal and Denominator Net*: Lets you select the numerator instance terminal and the denominator inet from the schematic.
- *Numerator Terminal and +-Denominator Nets*: Lets you select the numerator instance terminal and the differential denominator instance terminals from the schematic.

## For AC

1. Choose *Results – Direct Plot – Main Form*. The Direct Plot Form appears.
2. Select *ac* in the *Analysis* section.

   The Direct Plot Form is updated, as shown below:





© 2023 Cadence Design Systems, Inc.

3. Specify the *Plotting Mode*. You can specify:

   - *Append* to append the new signal to the current graph.

     **Note:** The *Append* mode is not recommended for plots with different scales and units for the X axis. It can give strange results because *Virtuoso Visualization and Analysis XL* opens a new subwindow to plot any data that does not match the units and range of the existing traces.

   - *Replace* to replace the current graph with the new signal. This is the default option.

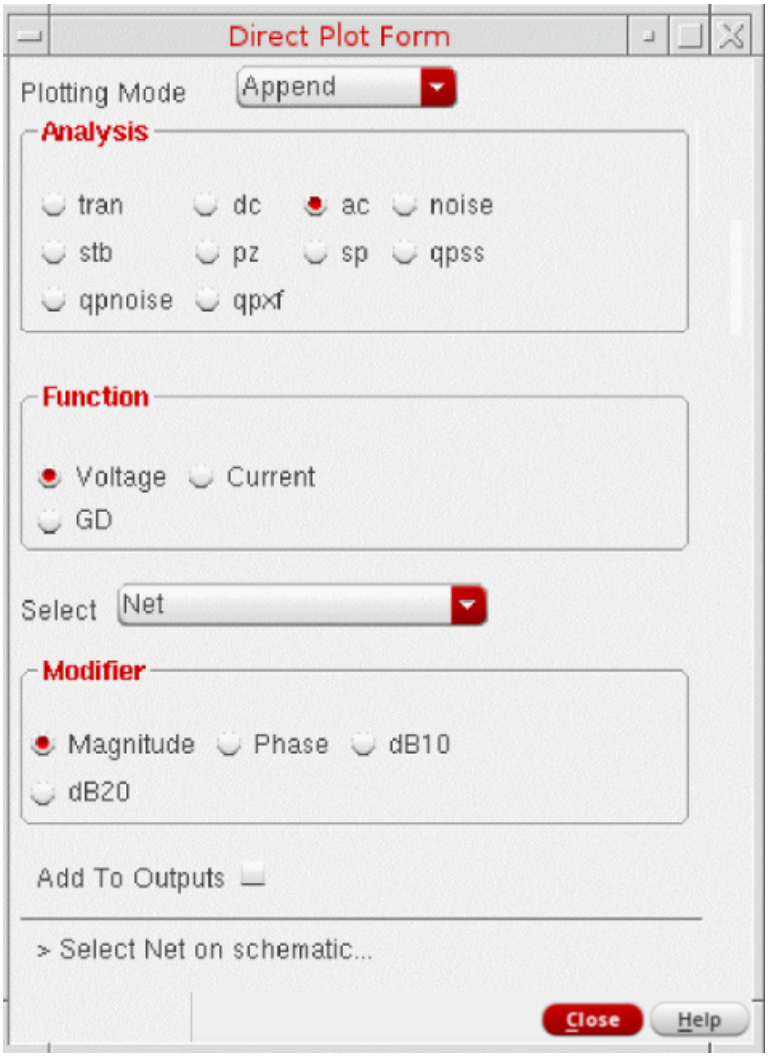
   - *New SubWin* to plot the signal in a new subwindow.

   - *New Win* to plot the signal in a new window.

4. The functions that are available are: *Voltage*, *Current*, *Current Ratio* and *GD*. Based on the selected function and available data, the form changes dynamically to display the applicable options.


© 2023 Cadence Design Systems, Inc.

5. Choose the nets and terminals to plot. You can select *Terminal, Net*, *Differential Nets* or *Instance with 2 Terminals* in the schematic to display the graphs for the selected analysis type.



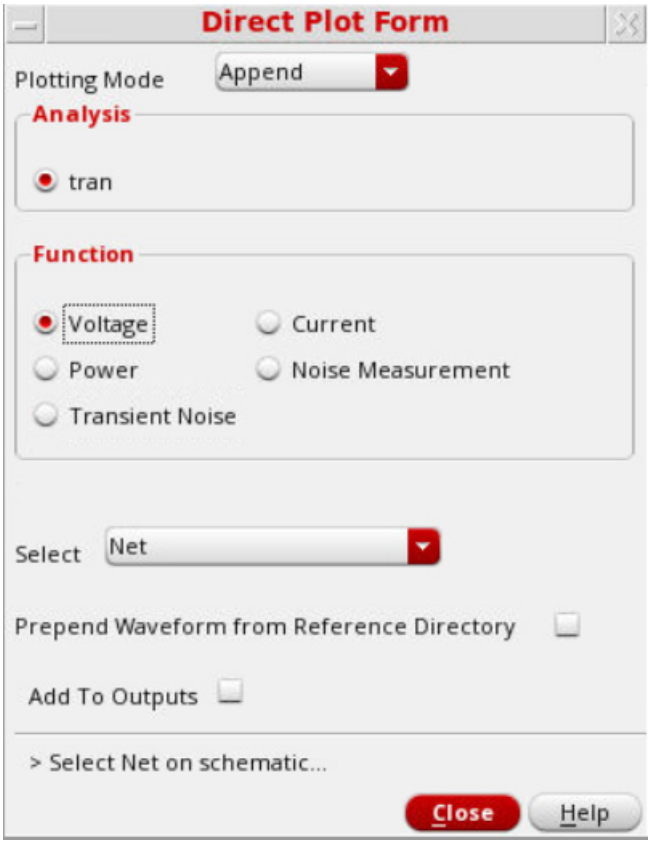

6. Choose the appropriate *Modifier* to specify the data or plot format.
7. Enable *Add To Output*s to add expressions for the results to the outputs section and plot in the mode that you selected.

## For Transient Results

1. Choose *Results – Direct Plot – Main Form*. The Direct Plot Form appears.
2. Select *tran* in the *Analysis* section.


© 2023 Cadence Design Systems, Inc.

The Direct Plot Form is updated, as shown below:




3. Specify the *Plotting Mode*. You can specify:
   - *Append* to append the new signal to the current graph.
   - *Replace* to replace the current graph with the new signal. This is the default option.
   - *New SubWin* to plot the signal in a new subwindow.
   - *New Win* to plot the signal in a new window.
4. The functions that are available are: *Voltage*, *Current*, *Power*, *Noise Measurement*, and *Transient Noise*.

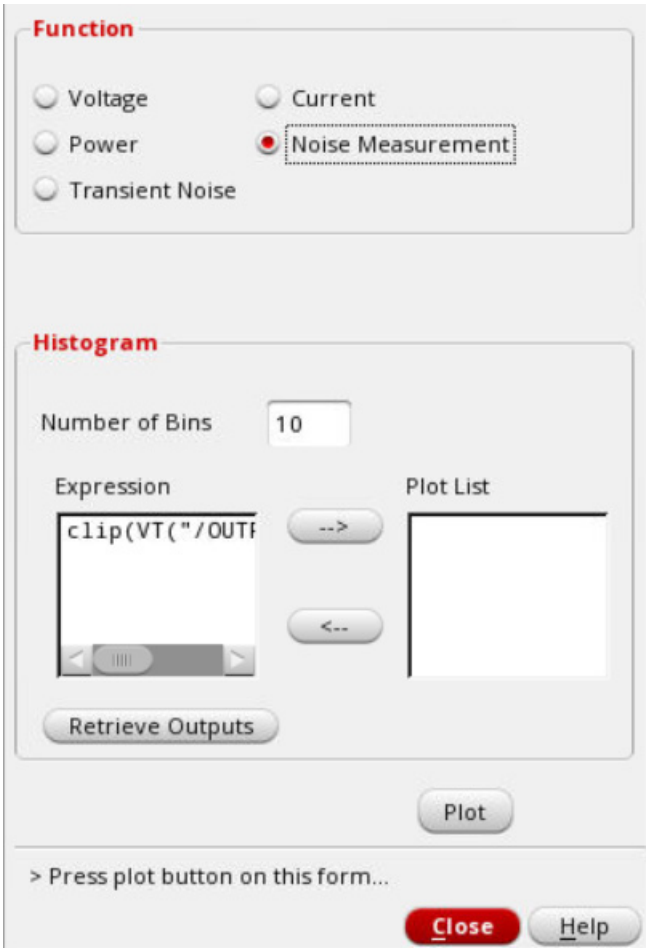
   Based on the selected function and available data, the form changes dynamically to display the applicable options.

   - For the *Voltage*, *Current*, and *Power* functions, choose the net or terminal to plot. These functions are the same as those available in the *Direct Plot* form For DC analysis.


© 2023 Cadence Design Systems, Inc.

The *Prepend Waveform from Reference Directory* option can be used for appending multiple checkpoint/restart transient waveforms together to enable you to view complete waveforms. Specify the reference results directory(s) in the field. The signal you choose in your direct plot will then be accessed from the reference directory.

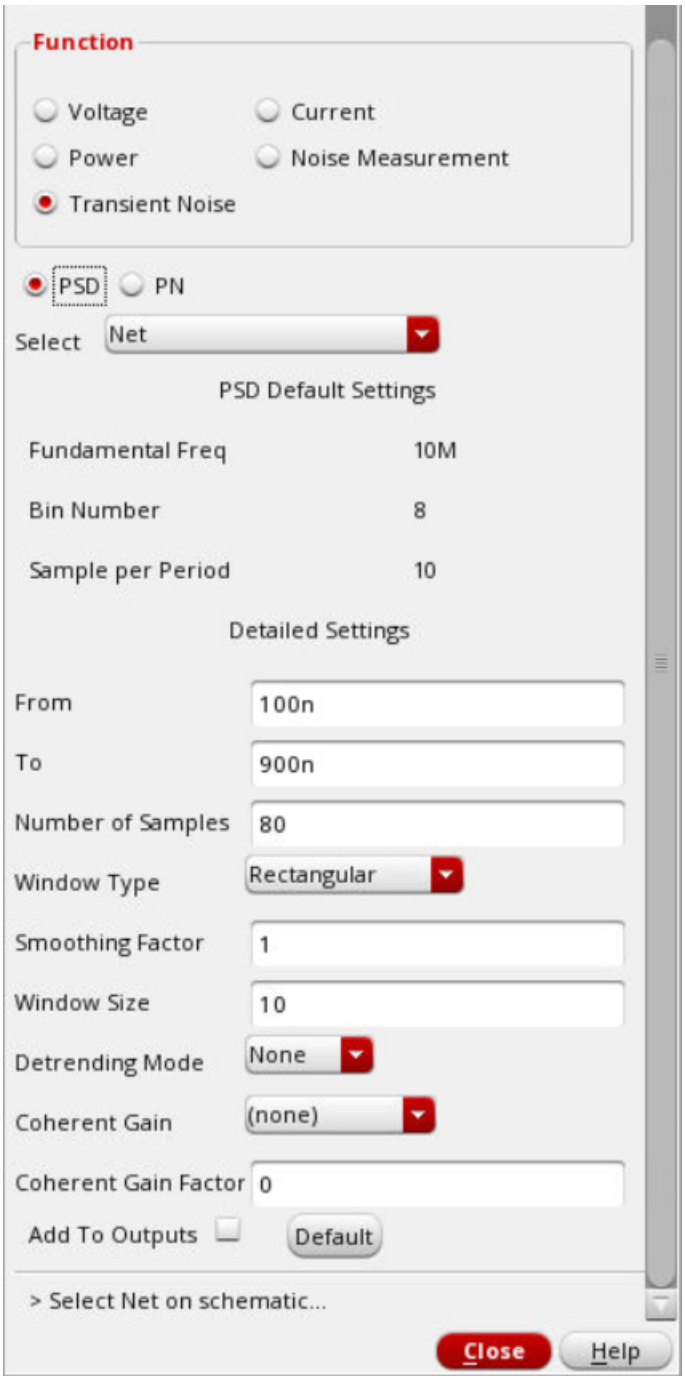
- For the *Noise Measurement* function, specify the values of the options that appear in the *Histogram* section in the form.




See *Histogram Plots for Transient Noise Analysis* for more information.


© 2023 Cadence Design Systems, Inc.

- For the *Transient Noise* function, specify the values of the following additional options that appear in the form and choose the net or instance to plot.





© 2023 Cadence Design Systems, Inc.

The *PSD* function generates a PSD waveform for the specified net or instance, and the *PN* function generates a waveform of the transient phase noise for the specified net or instance.

| Field | Description |
|---|---|
| **PSD** | |
| *From* | Starting time for the spectral analysis interval. |
| *To* | Ending time for the spectral analysis interval. |
| *Number of Samples* | Number of time domain points to be used. |
| *Window Type* | Window you want to use. If you select `Kaiser`, type in a value for the Kaiser smoothing factor. The smoothing factor must be in the range 0 <= factor <= 15, where 0 is the same as using a rectangular window.<br>Valid values: `Blackman`, `Cosine2`, `Cosine4`, `ExtCosBell`, `HalfCycleSine`, `HalfCycleSine3`, `HalfCycleSine6`, `Hamming`, `Hanning`, `Kaiser`, `Parzen`, `Rectangular`, and `Triangular`<br>Default value: `Hanning` |
| *Smoothing Factor* | Controls the output accuracy. This argument applies only to the `Kaiser` window type. It is applicable to the Kaiser window only. The Smoothing Factor accepts values from `0` to `15`. The value `0` implies no smoothing and is equivalent to a rectangular window.<br>Default value: `1` |
| *Window Size* | Number of frequency domain points to use in the Fourier analysis. A larger window size results in an expectation operation over fewer samples, which leads to larger variations in the power spectral density. A small window size can smear out sharp steps in the power spectral density that might be present.<br>Default value: `256` |


© 2023 Cadence Design Systems, Inc.

| Field | Description |
|---|---|
| *Detrending Mode* | Specifies the trend mode. The `psd` function works by applying a moving windowed FFT to time-series data. If there is a deterministic trend to the underlying data, you may want to remove the trend before performing the spectral analysis. For example, consider analyzing phase noise in a VCO model. Without the noise, the phase increases more or less linearly with time, so it is appropriate to set the detrending mode to `Linear`. To subtract an average value, set the detrending mode to `Mean`. Where the spectrum of raw data is desired, set the detrending mode to `None`.<br><br>Valid values: `None`, `Mean`, and `Linear`<br><br>Default value: `None` |
| *Coherent Gain* | The coherent gain of a window is the zero frequency gain (or the DC gain) of the window. It is calculated by normalizing the maximum amplitude of the window to one and then summing the values of the window amplitudes over the duration of the window. The result is then divided by the length of the window (that is, the number of samples). When you do a dft, the applied Window Type (aside from the rectangular type) changes the signal amplitude. Applying a coherent gain factor is a way to get consistent results regardless of the window type.<br><br>Valid Values: `(none)`, `(default)`, `(magnitude)`, `(dB20)`, and `(dB10)`<br><br>Default value: `(default)` |
| *Coherent Gain Factor* | This is the scaling factor. A non-zero value that scales the power spectral density by 1/(Coherent Gain).<br><br>Valid values: 0 < coherent_gain_factor < 1. You can use `1` if you do not want this scaling factor to be used.<br><br>Default value: `0` |
| **PN** | |
| *From* | Starting time for the spectral analysis interval. |
| *To* | Ending time for the spectral analysis interval. |


© 2023 Cadence Design Systems, Inc.

| Field | Description |
|---|---|
| *Cross Type* | Points at which the curves of the waveform intersect with the threshold. While intersecting, the curve may be either rising (rising) or falling (falling). The default value is rising. |
| *Threshold* | Threshold value against which the phase noise is to be calculated.<br><br>**Note:** A threshold tolerance of 0.1*stddev(signal) is used when detecting crossings. |
| *Tnom* | Nominal time period of the input waveform.<br><br>If left empty, the `pn` function takes the value of this field to be nil automatically.<br><br>Default value: `nil` |
| *Window Type* | Window you want to use.<br><br>Valid values: `Blackman`, `Cosine2`, `Cosine4`, `ExtCosBell`, `HalfCycleSine`, `HalfCycleSine3`, `HalfCycleSine6`, `Hamming`, `Hanning`, `Kaiser`, `Parzen`, `Rectangular`, and `Triangular`<br><br>Default value: `Rectangular` |
| *Smoothing Factor* | Smoothing factor applicable to only Kaiser window. The Smoothing Factor field accepts values from `0` to `15`. The value `0` implies no smoothing and is equivalent to a rectangular window.<br><br>Default value: `1` |
| *Window Size* | Number of frequency domain points to be used while calculating the power spectral density. A larger window size results in an expected operation over fewer samples, which can result in larger variations in the phase noise. A small window size can smear out sharp steps in the phase noise that might be present.<br><br>Default value: `256` |
| *Detrending Mode* | Determines the expected trend for the underlying data while calculating the power spectral density.<br><br>Default value: `None` |

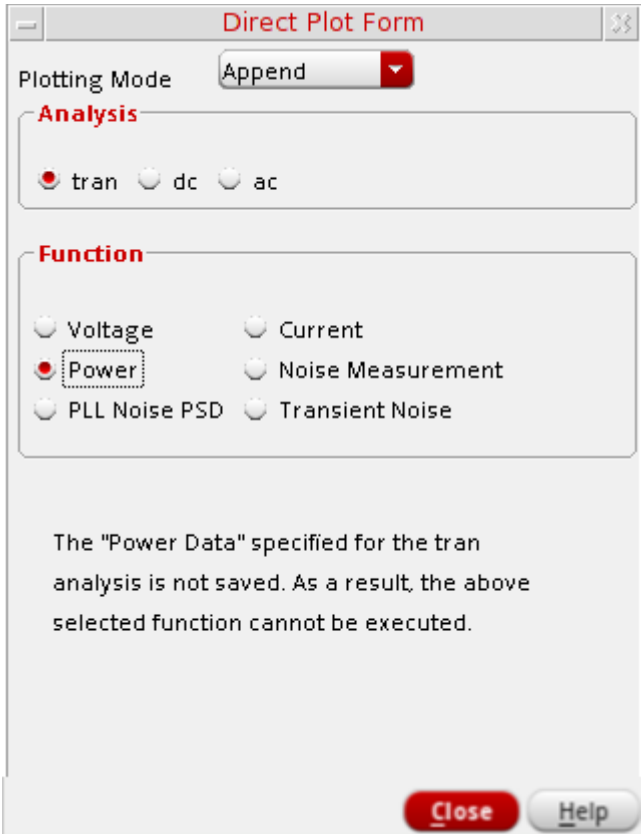

© 2023 Cadence Design Systems, Inc.

| Field | Description |
|---|---|
| *Coherent Gain* | The coherent gain of a window is the zero frequency gain (or the DC gain) of the window. It is calculated by normalizing the maximum amplitude of the window to one and then summing the values of the window amplitudes over the duration of the window. The result is then divided by the length of the window (that is, the number of samples). When you do a dft, the applied Window Type (aside from the rectangular type) changes the signal amplitude. Applying a coherent gain factor is a way to get consistent results regardless of the window type.<br><br>Valid Values: `(none)`, `(default)`, `(magnitude)`, `(dB20)`, and `(dB10)`<br><br>Default value: `(default)` |
| *Coherent Gain Factor* | This is the scaling factor. A non-zero value that scales the power spectral density by 1/(Coherent Gain).<br><br>Valid values: 0 < coherent_gain_factor < 1. You can use `1` if you do not want this scaling factor to be used. Default value: `0` |
| *Method Type* | Determines the algorithm you want to use to calculate the phase noise.<br><br>Valid values: `Absolute Jitter Method` or `Direct Power Spectral Density Method`.<br><br>Default value: `Absolute Jitter Method`.<br><br>**Note:** When the method type is selected as `Direct Power Spectral Density Method`, ensure that a valid nominal time period value is specified in the *Tnom* field, otherwise, an error message is displayed. |

Coherent GainFor more information about the options required to plot these waveforms, see the section PSD or PN in the *Virtuoso Visualization and Analysis XL User Guide*.


© 2023 Cadence Design Systems, Inc.

If no data is available for the selected function, a corresponding message is displayed in the form.




## For Noise Results

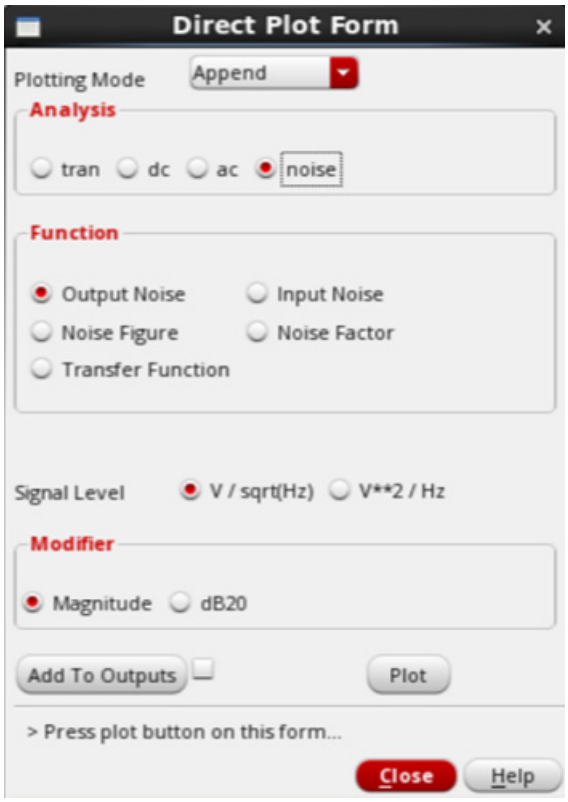
Noise figure is calculated by *Spectre* if a port is selected as the input source for noise analysis. If a port is not selected as the input, noise figure data is not available.

To plot the noise figure,

1. Choose *Results – Direct Plot – Main Form*. The Direct Plot Form appears.
2. Select *noise* in the *Analysis* section.


© 2023 Cadence Design Systems, Inc.

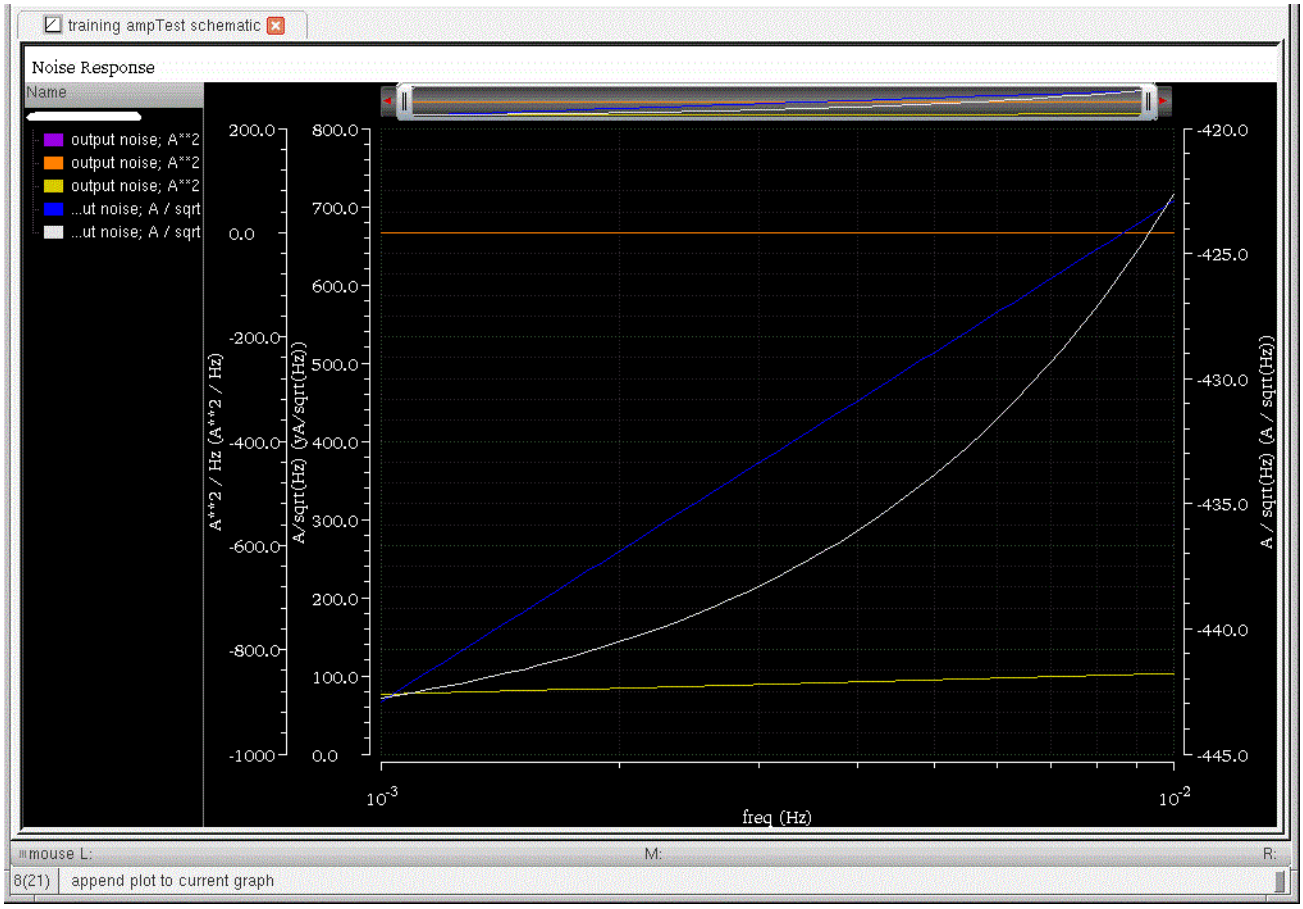
The Direct Plot Form is updated, as shown below:




3. Specify the *Plotting Mode*. You can specify:
   - *Append* to append the new signal to the current graph. This is the default option.
   - *Replace* to replace the current graph with the new signal.
   - *New SubWin* to plot the signal in a new subwindow.
   - *New Win* to plot the signal in a new window.
4. Select the noise parameter function that you want to plot from the *Function* group box. Based on the selected function and available data, the form changes dynamically to display the applicable options.
5. Select the *Signal Level* that you want to plot. By default, ADE Explorer plots the `VN2` signal in the *Virtuoso Visualization & Analysis XL* window. You can plot the `VN` signal by selecting the *V / sqrt(Hz)* radio button from the *Direct Plot* form.
6. Choose the appropriate *Modifier* to specify the data or plot format. *Magnitude* radio button (the default setting) plots the magnitude of the selected signal and the *dB20* radio button plots the magnitude in dB.


© 2023 Cadence Design Systems, Inc.

7. Select *Add To Outputs* check box to add expressions for the results to the outputs section and plot in the mode that you selected.
8. Click *Plot* to view the results in the *Virtuoso Visualization & Analysis XL* window.




9. Click *OK*.

**Note:** The mathematical noise-figure expression is where:

$$NF = 10\log\left(\frac{VN2*|Vin|}{C*|Vout|*\mathrm{R}}\right)$$

| | |
|---|---|
| *VN2* = | The noise voltage |
| *Vin* = | The voltage at the input node |
| *Vout* = | The voltage at the output node |


© 2023 Cadence Design Systems, Inc.

| | |
|---|---|
| *R =* | The source resistor value |
| *C =* | 1.61e-20 ≅ 4kT∆f |
| with | |
| *T =* | 291.5K and ∆f = 1 |

## For Noise Separation

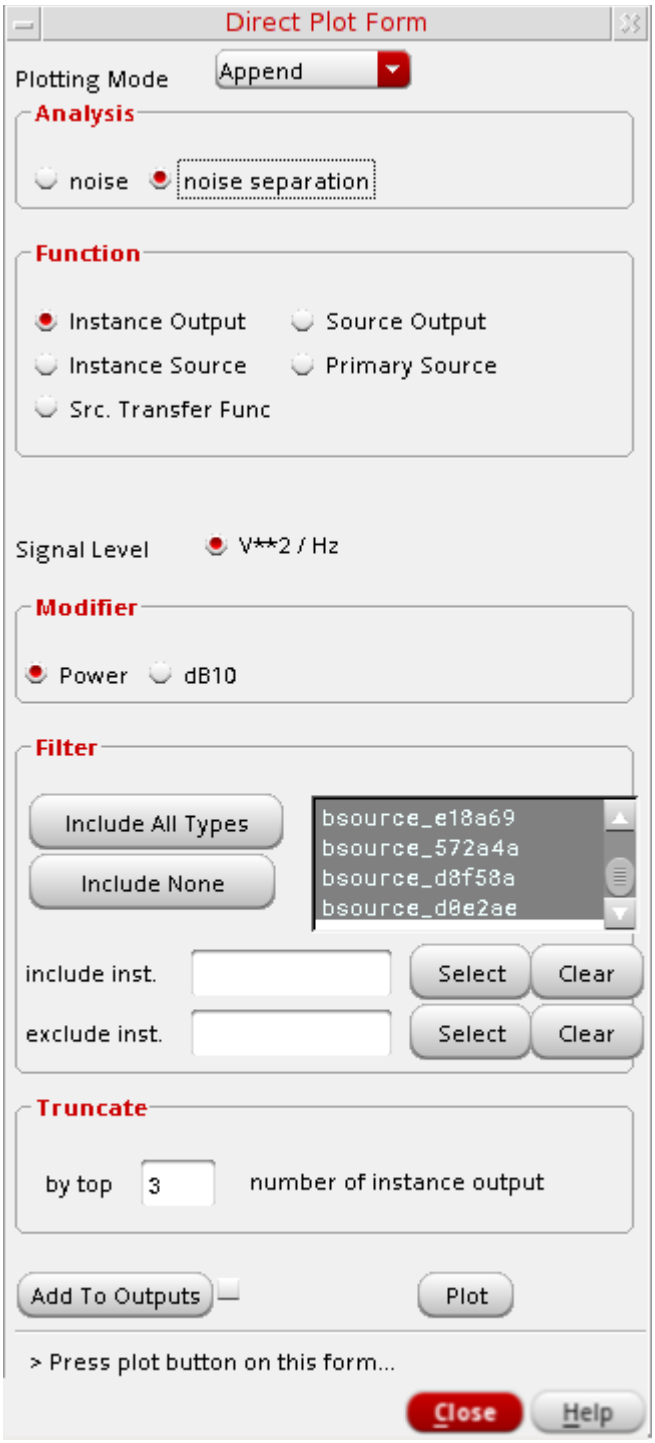
In addition to the total output noise, individual noise contributions can be plotted if *Noise Separation* is enabled in the Choosing Analyses form for Noise Analysis.

To plot the noise separation results,

1. Choose *Results – Direct Plot – Main Form*. The Direct Plot Form appears.
2. Select *noise separation* in the *Analysis* section.


© 2023 Cadence Design Systems, Inc.

The Direct Plot Form is updated, as shown below:




3. Specify the *Plotting Mode*. You can select one of the following modes:


© 2023 Cadence Design Systems, Inc.

- *Append* — Select this to append the new signal to the current graph. This is selected by default.
- *Replace* — Select this to replace the current graph with the new signal.
- *New SubWin* — Select this to plot the signal in a new subwindow.
- *New Win* — Select this to plot the signal in a new window.

4. In the *Function* section, select the noise separation function that you want to plot. You can select one of the following functions:
   - *Instance Output* — Plots the noise contribution of the instances.
   - *Instance Source* — Plots the noise sources of the instances.
   - *Source Output* — Plots the noise contribution of the primary noise sources.
   - *Primary Source* — Plots the primary noise sources.
   - *Src. Transfer Func* — Plots the square of the transfer function square of the primary noise sources.

   Based on the selected function and available data, the form changes dynamically to display the applicable options.
5. Based on the selected function, the signal level is also detected automatically.
6. Choose the appropriate modifier to specify the data or plot format. *Power* (the default setting) plots the power of the selected signal. *dB10* plots the magnitude in dB.
7. In the *Filter* section, choose filtering details.

   The *Include All Types* option should be selected. But, if you just want the noise from specific types of noise generators, you can select them from the list box. Also, specify the instances to be included or excluded particular in your summary.
8. In the *Truncate* section, choose truncation details to shorten your summary.

   You can shorten your summary by specifying how many of the highest contributors to include in the summary.
9. Select the *Add To Outputs* check box to add expressions for the results to the outputs section and plot in the mode that you selected.
10. Click *Plot* to view the results in the *Virtuoso Visualization & Analysis XL* window.
11. Click *OK* to close the form.

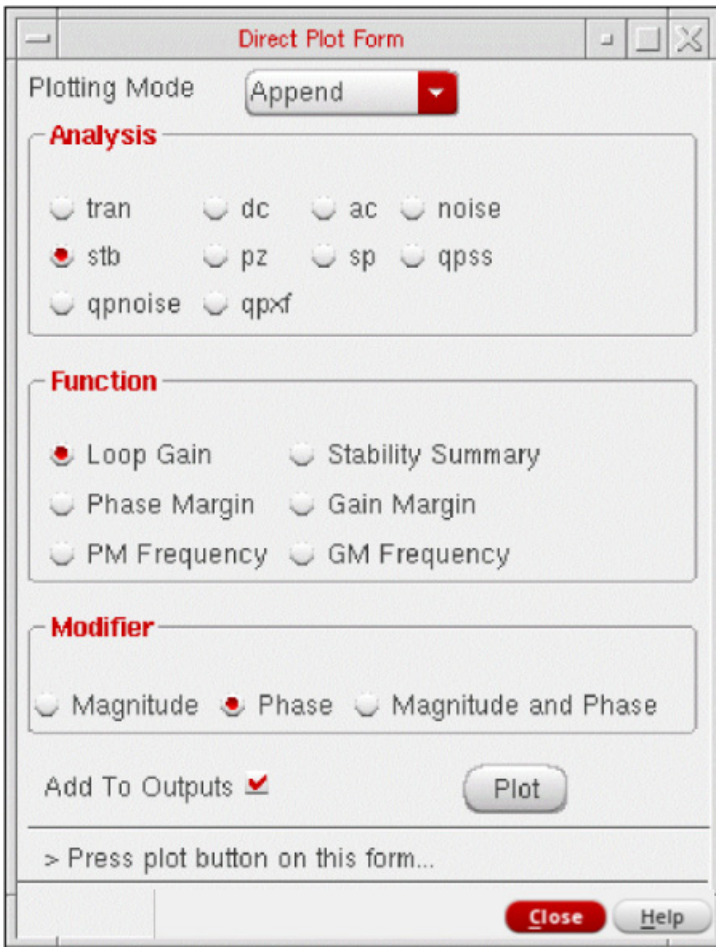

© 2023 Cadence Design Systems, Inc.

## For Stability Results

1. Choose *Results – Direct Plot – Main Form*. The Direct Plot Form appears.
2. Select *stb* in the *Analysis* section.

   The Direct Plot Form is updated, as shown below:

   .

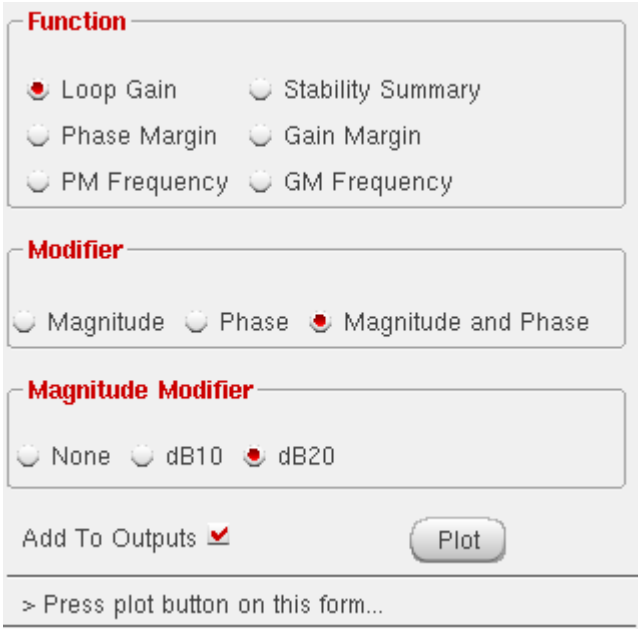



3. Specify the *Plotting Mode*. You can specify:
   - *Append* to append the new signal to the current graph.
   - *Replace* to replace the current graph with the new signal. This is the default option.
   - *New SubWin* to plot the signal in a new subwindow.
   - *New Win* to plot the signal in a new window.


© 2023 Cadence Design Systems, Inc.

4. The functions that are available are: *Loop Gain*, *Stability Summary*, *Phase Margin*, *Gain Margin*, *PM Frequency* and *GM Frequency*. Based on the selected function and available data, the form changes dynamically to display the applicable options.

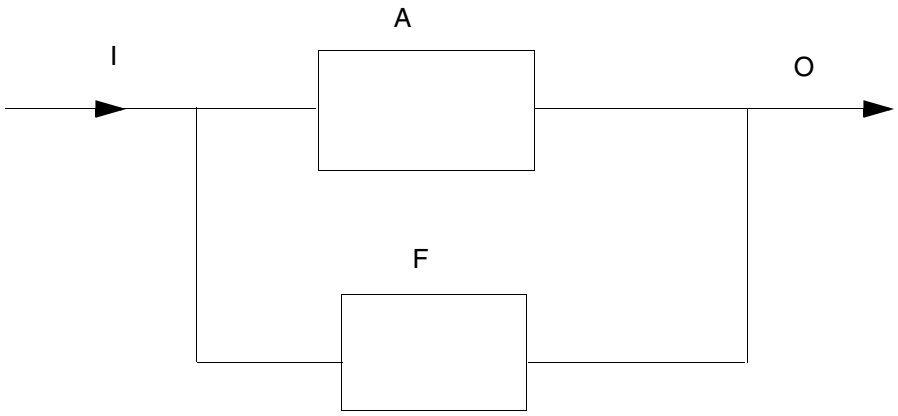
   **a.** When you select *Loop Gain*, the form re-displays to show the *Modifier* section.The loop gain output is a complex waveform and you can select it to plot *Magnitude*, *Phase* or both (*Magnitude and Phase*). Whenever you choose to plot *Magnitude*, the *Magnitude Modifier* section appears on the form. You can select *None*, *dB10* or *dB20*, as needed. Whenever you plot both the magnitude and phase, the graph window changes to the strip mode. It reverts back to the composite mode for other plot operations.




   **Note:** There is a difference between the ADE Explorer and Spectre definition of loop gain.


© 2023 Cadence Design Systems, Inc.

For the feedback circuit shown below:




The closed loop gain is defined as:

$$\frac{O}{I} = \frac{A}{1 - AF}$$

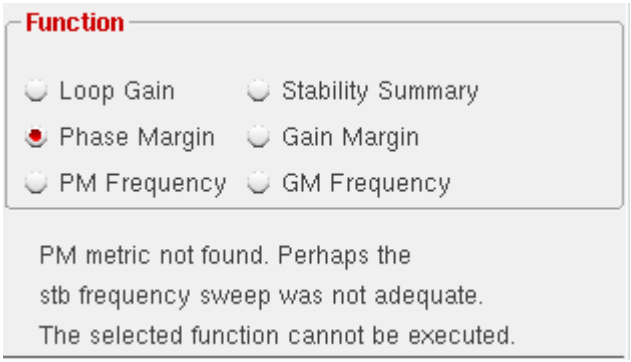
The Spectre output defines loop gain as the product **AF**, while others (like the ADE Explorer Calculator `phaseMargin` and `gainMargin` functions) define **-AF** as the loopGain. Therefore, to obtain the same results from ADE Explorer, you need to negate the Spectre's loopGain as illustrated below:

```
gainMargin( -1 * getData( "loopGain" ?result "stb" ), 1)
phaseMargin( -1 * getData( "loopGain" ?result "stb" ) )
```

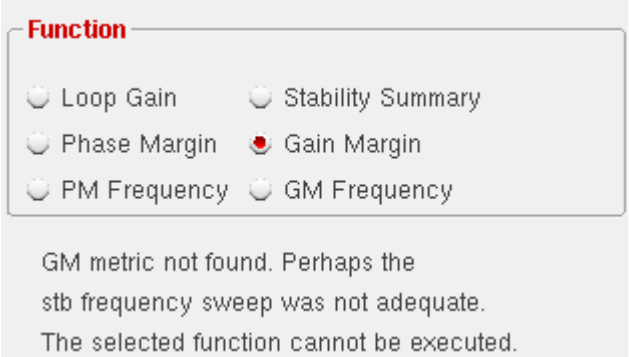
**b.** *Phase Margin*, *Gain Margin*, *PM Frequency* and *GM Frequency* constitute the margin data. This information is calculated from the loop gain data for the circuit. The information is only available when frequency is swept in the stability analysis and the


© 2023 Cadence Design Systems, Inc.

swept range is sufficient to calculate the values. When the selected margin data is scalar the values are displayed on the form itself.




When the swept frequency range is not sufficient to calculate the selected margin data an error is reported in the *Direct Plot Form* and the *Plot* and *Add to Outputs* button are not available.

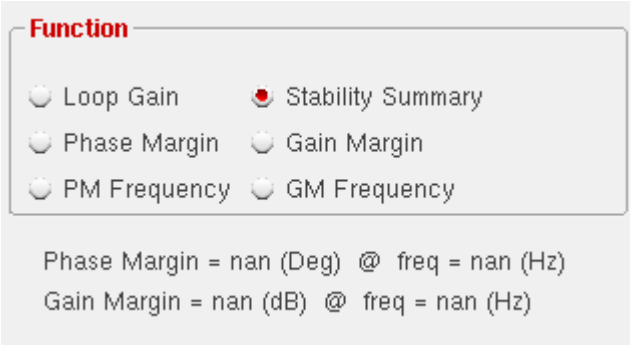



When frequency is not swept in the stability analysis and you choose any of the margin data functions an error is reported in the *Direct Plot Form* and the P*lot* and *Add to Outputs* button are not available.

**c.** Selecting *Stability Summary* displays all the margin data collectively on the form, when the data is scalar. You do not have the facility to plot or add the four outputs


© 2023 Cadence Design Systems, Inc.

when this function is chosen. Use the individual margin data function for this operation.




When frequency was not swept or the margin data is not scalar an appropriate error is reported in the *Direct Plot Form* and the P*lot* and *Add to Outputs* button are not available.

5. Enable *Add To Output*s and plot in the mode that you selected.

   **Note:** This option makes no checks for duplication in outputs.

All other parts of the *Direct Plot* form work the same way as they do for other analyses. Refer to the *Spectre Circuit Simulator RF Analysis User Guide* for details.

This form handles parametric (family) data. The *Loop Gain* would be a set of curves for family data. Similarly for non-parametric data, *Phase Margin* and *Gain Margin* will be scalars. A horizontal straight line will be plotted for them.

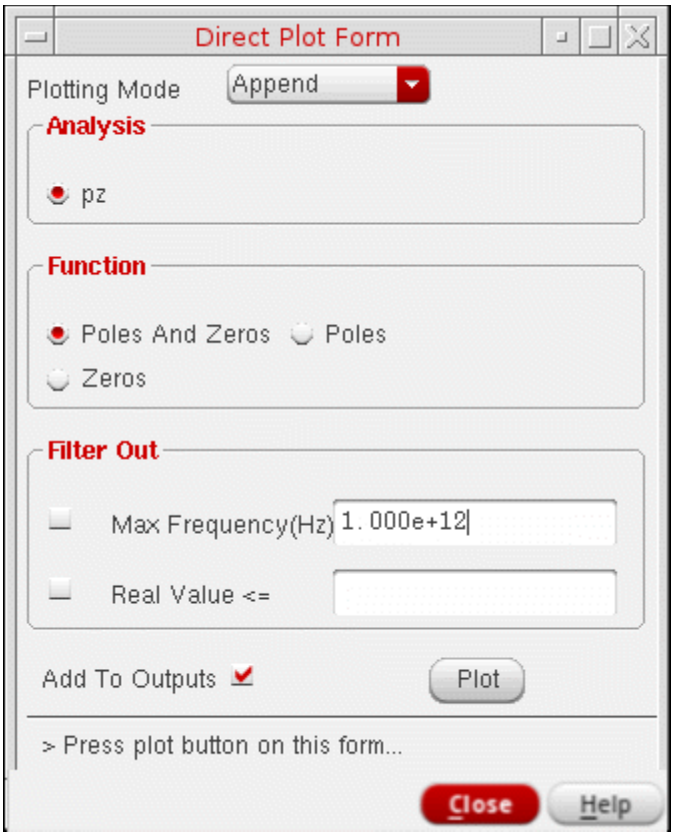
## For Pole Zero Results

Once you run a simulation for *Pole Zero* analysis, you can use the *Direct Plot* main form to view the poles and zeros plotted on the real/imaginary plane.

1. To open the Direct Plot Form, select *Results – Direct Plot – Main Form*.
2. Select *pz* in the *Analysis* section.


© 2023 Cadence Design Systems, Inc.

The Direct Plot Form is updated to display the applicable functions and options:




3. Select the option, *Poles* if you want to plot only poles, *Zeros* if you want to plot only zeros and *Poles and Zeros* if you want to plot both poles and zeros.

   **Note:** By default, the option *Poles and Zeros* is selected.

4. Set the required options in the *Filter Out* section and click *OK*. This section provides a combination of filtering mechanisms that you can select in order to plot the poles and zeros. These are:

   - *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 that specified in the *fmax* field in the *Pole-Zero Options* form. Only poles and zeros whose magnitudes exceed the frequency value specified are filtered out.

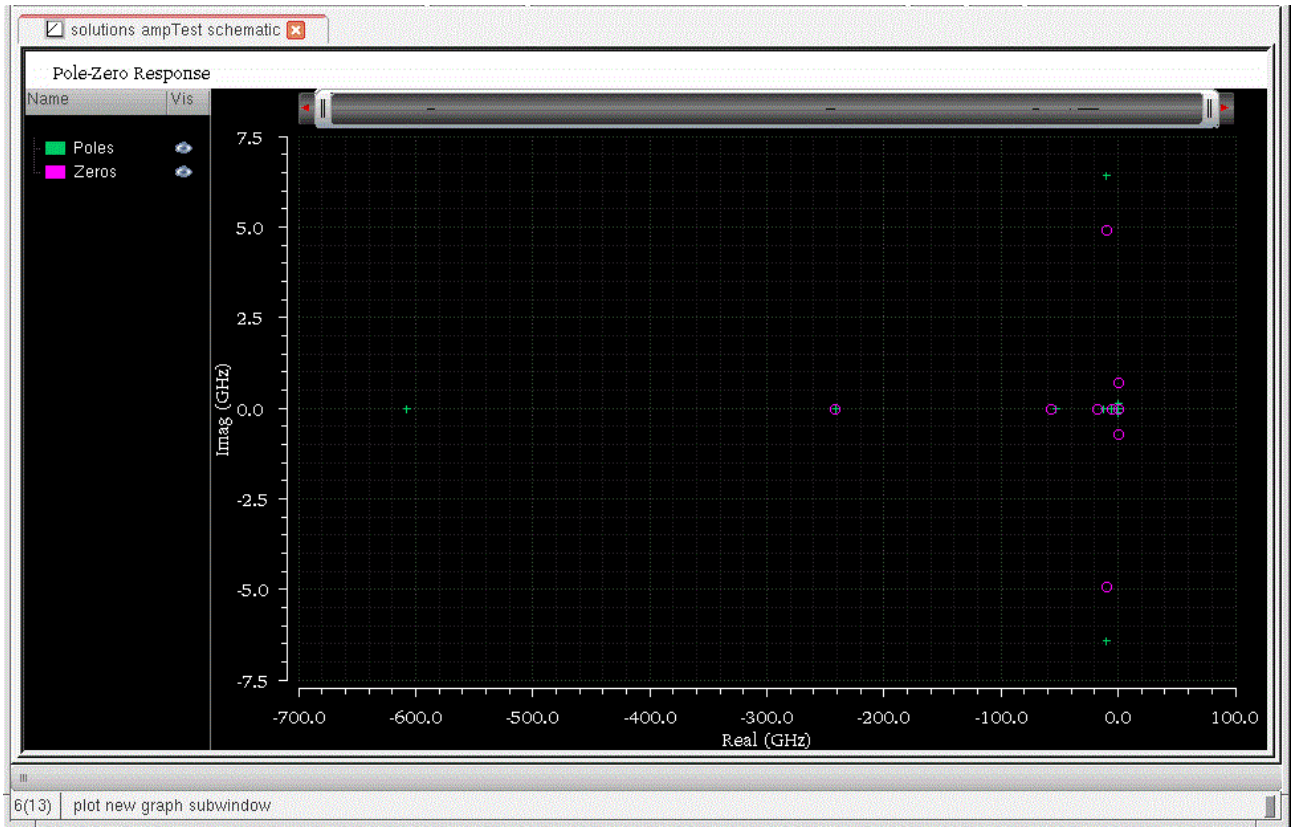
   - *Real Value:*
     This option enables you to specify the real part of the frequency. Only poles and zeros whose real values are less than or equal to the real value specified are filtered out.


© 2023 Cadence Design Systems, Inc.

**Note:** By default, no filtering is selected. You can set the filtering criteria once you specify either poles or zeros or both to be plotted.

5. Enable *Add To Outputs* to add expressions for the results to the Outputs Setup and plot in the mode that you selected.

6. Click *Plot* to view the results in the *Graph Window*:




Poles and zeros are plotted in *scatter* mode. This implies that poles and zeros are plotted individually but not connected. Poles are represented by the symbol **x** and zeros by the symbol **o**. The complex data is plotted with poles and zeros.

### Non-Swept Parameters

For the non-swept case, the result of Pole Zero analysis will be two waveform objects, one representing poles and another representing the zeros. The two wave objects are plotted in the same color however, *poles* will be represented by the symbol **x** and *zeros* by the symbol **o**.


© 2023 Cadence Design Systems, Inc.

### Swept Parameters

For swept parameter Pole Zero Analysis, it is possible to create the root-locus plot. Instead, the poles and zeros are plotted corresponding to each *Swept Parameter* value.

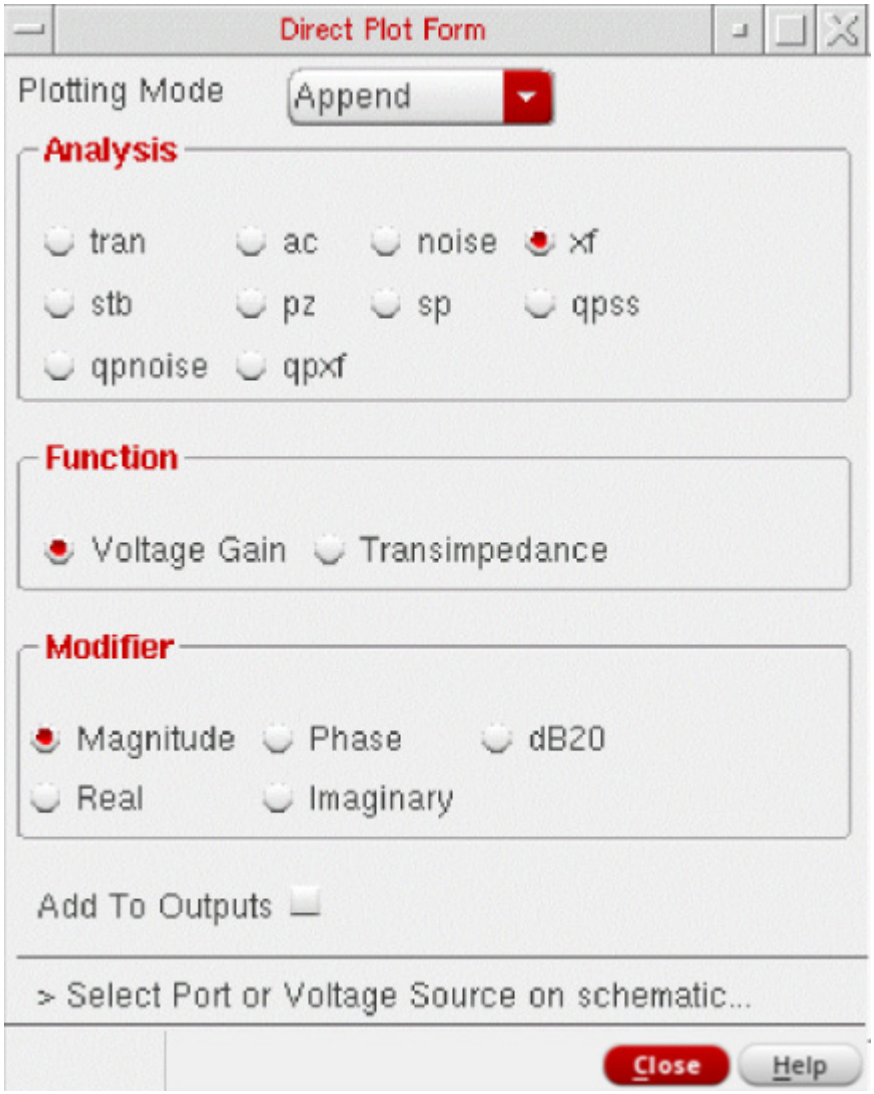
## For Transfer Function

To plot the transfer function,

1. Choose *Results – Direct Plot – Main Form*. The Direct Plot Form appears.
2. Select *xf* in the *Analysis* section.


© 2023 Cadence Design Systems, Inc.

The Direct Plot Form is updated, as shown below:




3. Specify the *Plotting Mode*. You can specify:
   - *Append* to append the new signal to the current graph.
   - *Replace* to replace the current graph with the new signal. This is the default option.
   - *New SubWin* to plot the signal in a new subwindow.
   - *New Win* to plot the signal in a new window.
4. Choose either *Voltage Gain* or *Transimpedance* if you selected output voltage for the transfer analysis, or *Current Gain* or *Transconductance* if you selected output current for the transfer analysis.


© 2023 Cadence Design Systems, Inc.

5. Specify the modifiers as needed.
6. Select either the instance or instance terminal in the schematic.

   The graph window redisplays, showing the new plot.

To replot with modifications, make changes to the specifications on the XF Results form and click *Replot*.

## For S-Parameter

A typical S-parameter direct plot shows a parameter function plotted against frequency, based on a pair of psin elements that define an input and an output circuit port.

You define S-parameter direct plots with the S-Parameter Results form. If the form does not offer a plot you want to generate (for example, plots of complex computed results), use the waveform calculator.

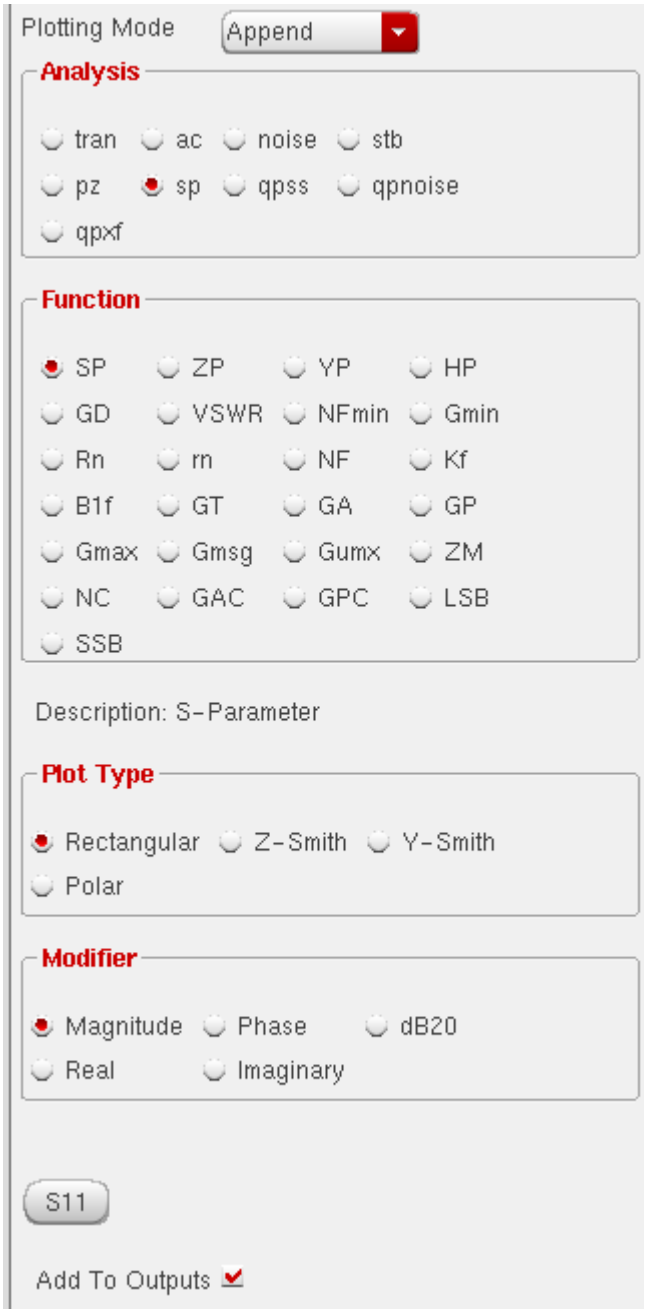
By default, the plots appear in the current Virtuoso Analog Design Environment graph window or subwindow. The current subwindow has a rectangle around its window number (in the upper-right corner). To use a different subwindow, select it before beginning the direct plot procedure. If no graph window or subwindow is open, this plot function automatically opens one.

1. To plot the results, choose *Results – Direct Plot – Main Form.*

   This command automatically opens a graph window (unless one is already open), the design schematic (unless it is already open), and the Direct Plot Form.
2. Select *sp* in the analysis section.


© 2023 Cadence Design Systems, Inc.

The Direct Plot Form is updated, as shown below:




For detailed information about the form, see S-Parameter Results.

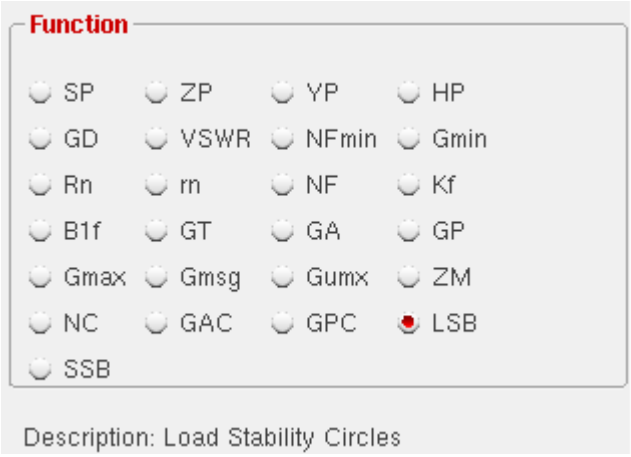
To plot S-parameter results,

1. Specify the *Plotting Mode*. You can specify:


© 2023 Cadence Design Systems, Inc.

- *Append* to append the new signal to the current graph.
- *Replace* to replace the current graph with the new signal. This is the default option.
- *New SubWin* to plot the signal in a new subwindow.
- *New Win* to plot the signal in a new window.

2. Click the radio button for the S-parameter or noise-parameter function you want to plot.




A brief description of the function appears below the buttons, and the bottom of the form changes to show options for the function.

**Note:** Some functions are defined only for two-port circuits. If you choose a function that is not available for your circuit data set, a warning message appears at the bottom of the form. Click a button on the figure for information about a function. If you need an equation that is not represented on the form, use the calculator to build, evaluate, and plot it.

3. Choose the appropriate *Plot Type* and *Modifier* to specify the plot type and the data or plot format.

4. Specify and draw the plot.

For S, Z, Y, or H parameters (shown as *SP*, *ZP*, *YP*, and *HP* on the form), generate plots for ports 1 through 3 by clicking the appropriate parameter button at the bottom of the form. To generate plots for any higher-numbered ports, use the cyclic fields beside the buttons to specify the output and incident ports. Then click the *S*, *Y*, *Z*, or *H* button that is next to the cyclic field to plot.

**Note:** For circuits with three or fewer ports, the form has no cyclic fields.


© 2023 Cadence Design Systems, Inc.

## S-Parameter Probing

Select *sp sprobe port* in the *Analysis* group box of the Direct Plot form.

The following fields are displayed.

1. From the *Select Sprobe* drop-down list, select an sprobe.
2. From the Select Port Direction drop-down list, select one of the following options.


© 2023 Cadence Design Systems, Inc.

- *Left*: uses the left port of the selected sprobe, specified through the Choosing Analyses form for sp analysis.
- *Right*: uses the right port of the selected sprobe, specified through the Choosing Analyses form for sp analysis.

3. Select a function from the *Function* group box.
4. Select a plot type from the *Plot Type* group box.
5. Select a modifier from the *Modifier* group box.
6. Click *Plot* to plot the wave.

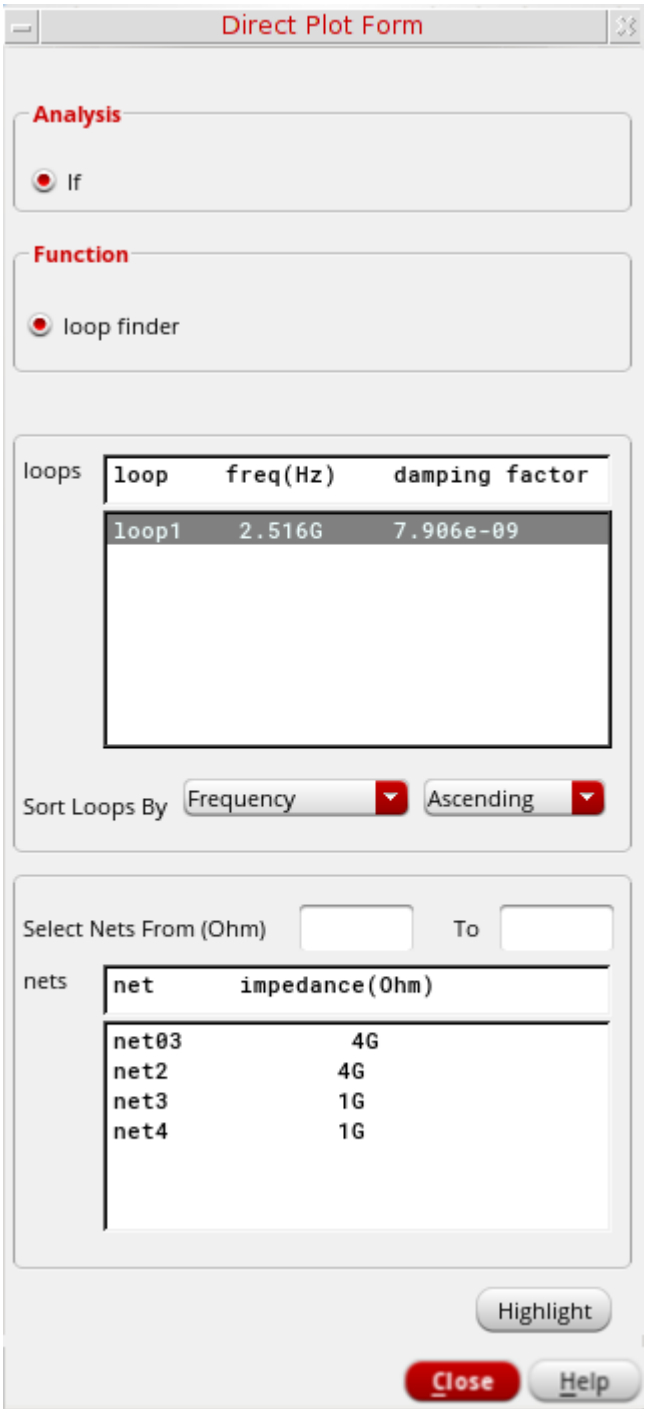
## For Loop Finder

To highlight the nets within a loop,

1. Choose *Results – Direct Plot – Main Form*. The Direct Plot Form appears.
2. Select *If* in the *Analysis* section.


© 2023 Cadence Design Systems, Inc.

The Direct Plot Form is updated, as shown below:




3. Choose a loop from the *loops* table. The circuit nodes (nets) within the selected loop are displayed in the *nets* table, along with their impedance value. If you want all the nets within a particular loop to be highlighted, double click the loop.


© 2023 Cadence Design Systems, Inc.

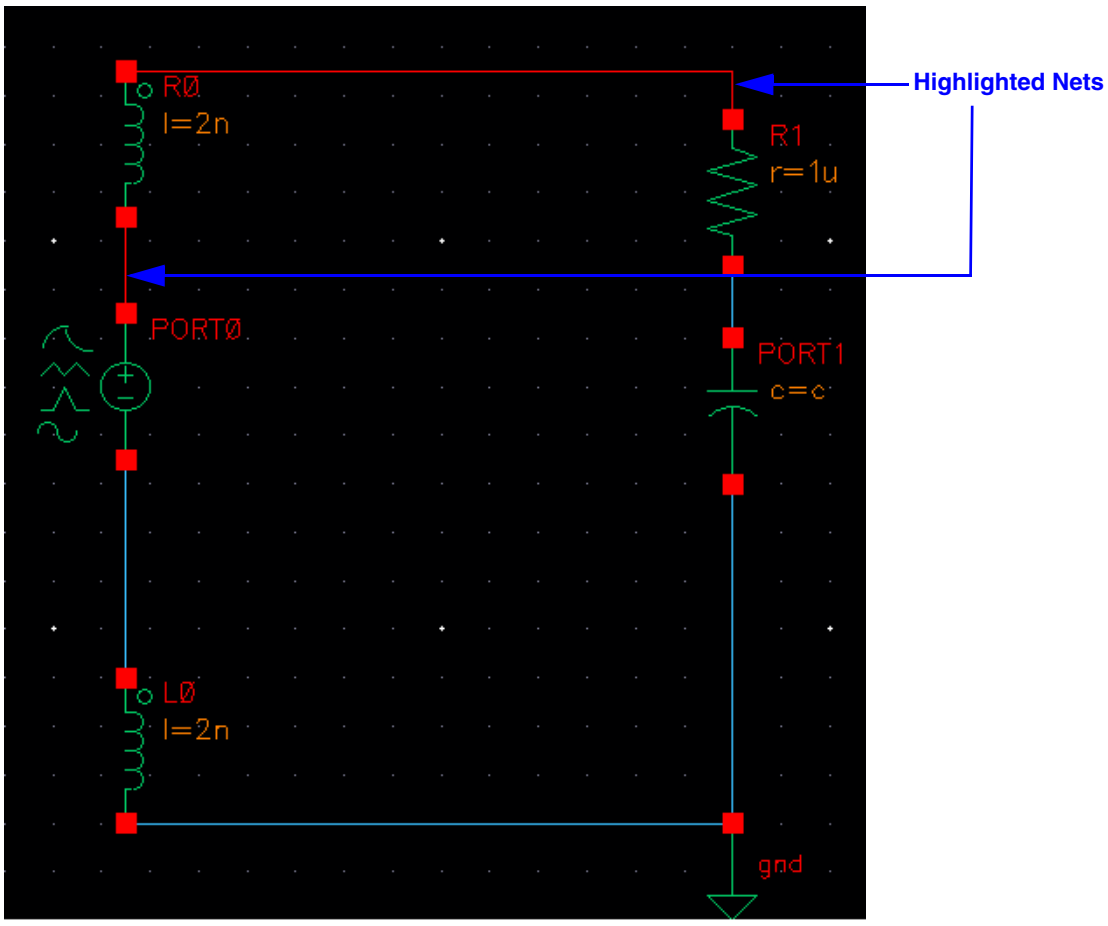
4. In the *Loops Sort By* field, you can specify a sorting criteria for the loops displayed in the *loops* table. You can sort the loops by frequency or damping factor, in ascending or descending order.

5. In the *Select Nets From (Ohm)* and *To* fields, specify the minimum and maximum impedance for the nets to be displayed. All the nets with an impedance lying outside the specified range are excluded from the *nets* table.

   If only the minimum impedance is specified in the *Select Nets From (Ohm)* field, all the nets with an impedance greater than the specified impedance are displayed. And, if only the maximum impedance is specified in the *To* field, all the nets with an impedance lower than the specified impedance are displayed.

6. Choose the nets to be highlighted from the *nets* list box. To select multiple nets, press the `Ctrl` key and select the nets, or (for contiguous selection) press the `Shift` key and select the nets.

7. Click *Highlight*. The selected nets are highlighted in the schematic, in red, as shown below:





© 2023 Cadence Design Systems, Inc.

8. Click *OK* to close the form.

# Printing Results

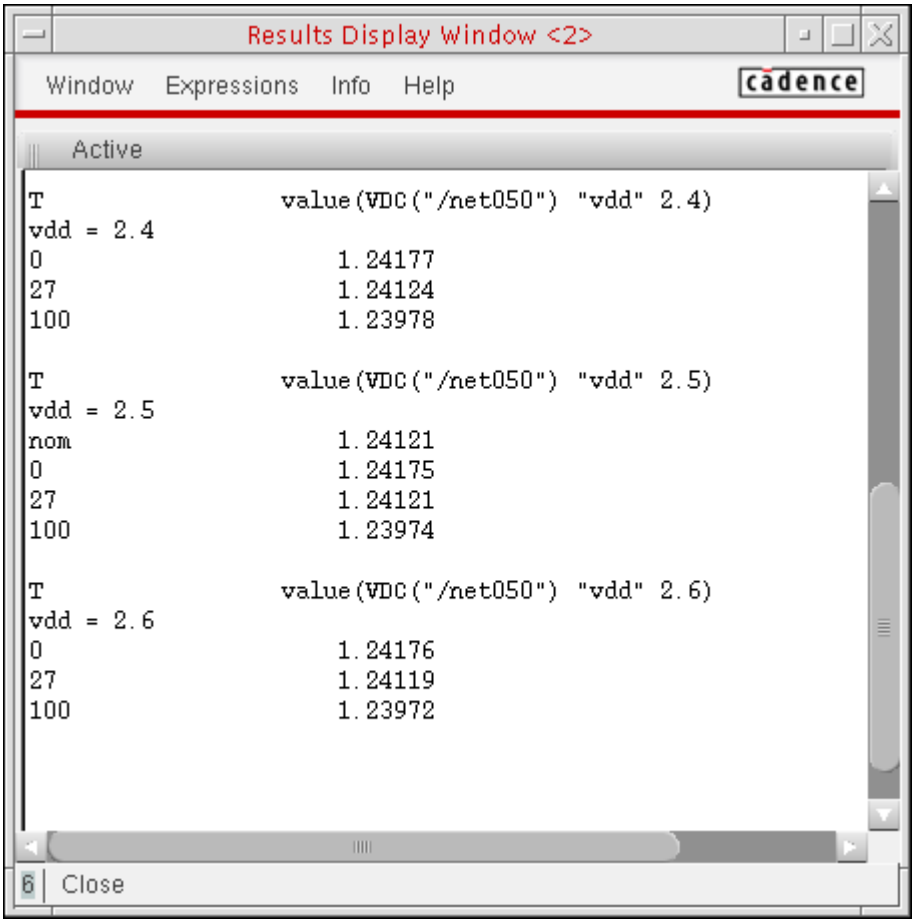
To print text results and reports to the Results Display Window, do the following:

- On the Results tab, right-click in a column and choose *Print*.

  A submenu appears to display only those functions that apply to the current results are available for selection.

- Choose a *Results – Print* menu command in the ADE Explorer window.

  The Results Display Window appears:




For guidance on using the Results Display Window to perform tasks, see the following sections.

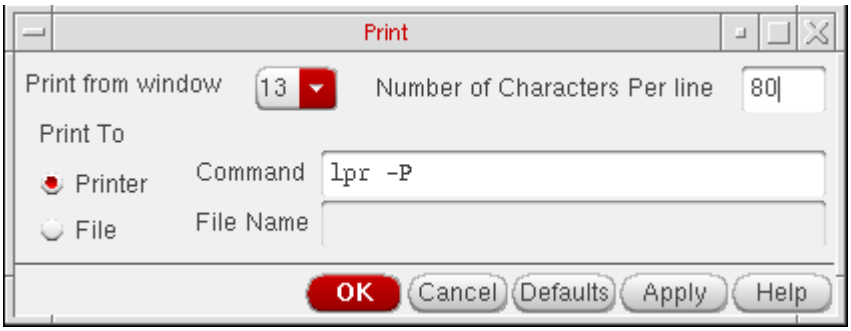

© 2023 Cadence Design Systems, Inc.

## Printing Results

To print the results in the Results Display Window either in hardcopy or to a file,

1. Choose *Window – Print*.

   The Print form appears.




2. Choose the correct window number from the *Print from window* cyclic field.

   This is the window containing the contents you want to print.

3. Type a value in the *Number of Characters Per Line* field.

4. Choose either the *Printer* or *File* radio button in the *Print To* field.

   You must type a filename if you choose *File*.

5. Click *OK*.

## Saving State from the Results Display Window

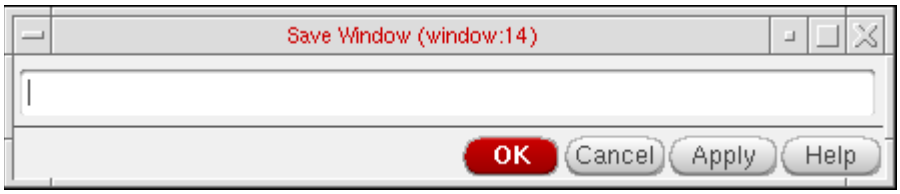
You can use *Save State* and *Load State* capability to save the current setup of display options for printing waveforms such as printing format, setting a printing range if the amount of data is too large, printing at a certain interval, and changing the order of the display. You can save the state of the window into a file. Later if you run another simulation and do *Load State*, the new data can be loaded back and displayed as you specified when you saved the state. *Save State* and *Load State* are applicable only to waveforms (that is, expressions that can evaluate to a waveform). If you print a single number, like a node voltage, these commands are disabled. You get a message stating this value is not a waveform and cannot be loaded back.

To save the contents and format of a Results Display Window,

1. Choose *Window – Save State*.


© 2023 Cadence Design Systems, Inc.

The Save Window form appears.



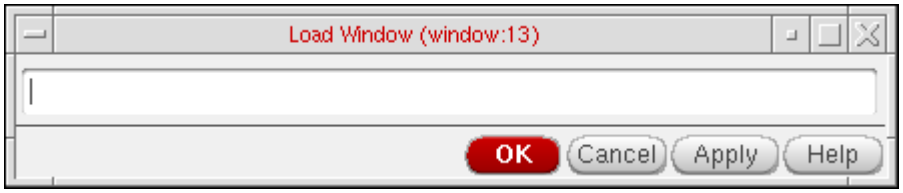

2. Type a filename in the field.
3. Click *OK*.

### Loading State in the Results Display Window

To load a window state that you previously saved,

1. Choose *Window – Load State*.

   The Load Window form appears.




2. Type the name of the saved file in the field.
3. Click *OK*.

### Updating Results in the Results Display Window

To update the Results Display Window with results from a new simulation,

➤ Choose *Window – Update Results*.

This updates the data using the current window setup. *Update Results* is applicable only to waveforms (that is, expressions that can evaluate to waveforms). If you print a single number, like a node voltage, this command is disabled.

### Making a Window Active

There is no limit to the number of Results Display Windows you can have open, but only one window is active at a time. All printouts go to the active window.


© 2023 Cadence Design Systems, Inc.

To make a window active,

➤ Choose *Window – Make Active* in the window that you want to make active.

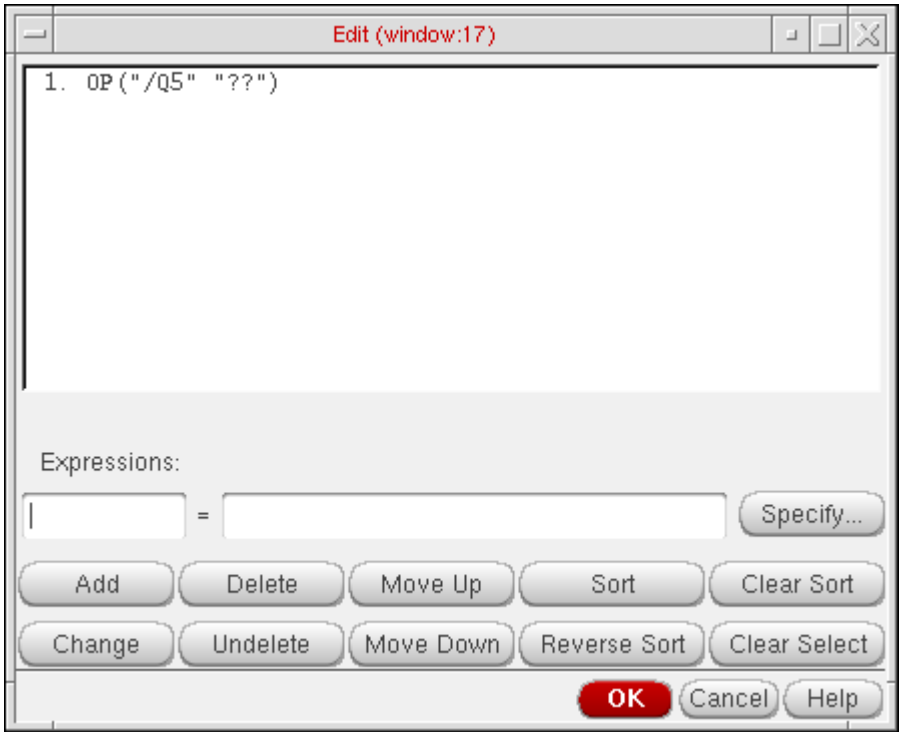
### Editing Expressions

You can edit any expressions that evaluate to waveforms (for example, DC operating parameters, model parameters, and transient operating parameters). If you print only one value, the edit menu choices are not available. The editing commands operate on only the last table in the active Results Display Window.

To edit expressions in the print window,

1. Choose *Expressions – Edit*.

   The Edit window appears.




2. Edit the expressions using the form buttons and fields.

**Note:** You should convert both numbers to the same type (that is, integer or float) before you compare them. For more information, see *Integer versus Floating-Point Division* in the *Arithmetic and Logical Expressions* chapter of the *Cadence SKILL Language User Guide*.


© 2023 Cadence Design Systems, Inc.

- *Expressions*- The field to the left of the equal sign shows the aliased name of the expression to the right. Naming expressions is optional and the field might be blank. If aliases are used, they are shown in the list box and the title line of the print window list box.
- *Specify*- Retrieves the expression in the calculator buffer and places it into the edit field.
- *Add*- Adds the expression in the edit field to the list box.
- *Change*- Replaces the selected expression in the list box with the one in the edit field.
- *Delete*- Deletes the selected expression from the list box.
- *Undelete*- Lets you undo the last delete.
- *Move Up*- Moves the display of the selected expression one step to the left. If the expression is already the leftmost, it is moved to the rightmost.
- *Move Down*- Moves the display of the selected expression one step to the right. If the expression is already the rightmost, it is moved to the leftmost.
- *Sort*- Sorts the selected expression so that the value increases down the column.
- *Reverse Sort*- Sorts the selected expression so that the value decreases down the column.
- *Clear Sort*- Reverts to the default order.
- *Clear Select*- Clears the selection in the list box. Also, you can clear entries from the list box by clicking on the entry while holding down the `Control` key.

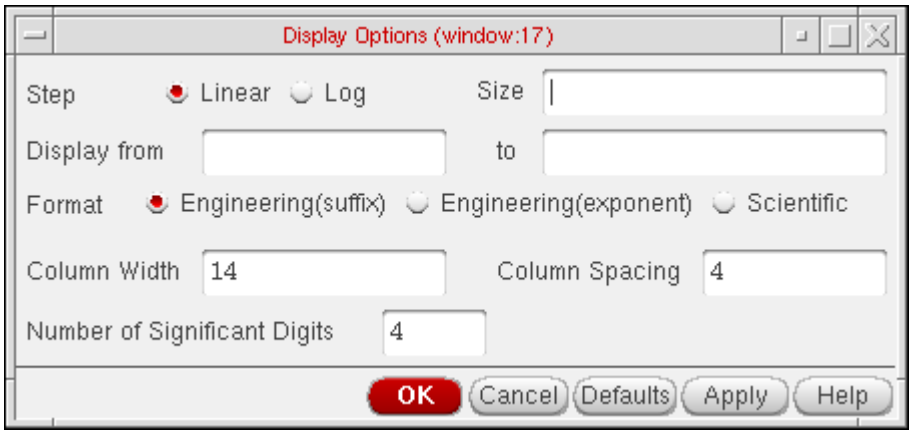
### Setting Display Options

To change the display options,

1. Choose *Expressions – Display Options*.


© 2023 Cadence Design Systems, Inc.

The Display Options form appears.




2. Type the values into the form and select a format.

**Note:** If the Results Display Window contains more than one type of results, the *Display Options* commands apply only to the last result (if the last result can evaluate to a waveform). After the data is edited, only the last result appears in the window. If you want to preserve the previous results, you can open a new Results Display Window and print the results to be edited in the new window.

- *Step size*- Specifies the interval for printing data.
- *Display from, to*- Specifies the range of data to print. If *from* is left blank, the data is printed from the beginning. If *to* is blank, the data is printed to the end. You can set the print range only after printing data.
- *Format*- Controls the format of the data printed. The possible formats are *Engineering Suffix* (default), *Engineering*, and *Scientific*. For example, you represent 0.0001 as 0.1m (engineering suffix), 0.1e-3 (engineering), or 1e-4 (scientific).
- *Linear/Log*- Specifies whether the scale used for step size is linear or logarithmic.
- *Column width/spacing*- Changes the number of characters allowed for column width and spacing. The default width is 14 characters. The allowed range is 4 to 20 characters. The default spacing is 4 and the allowed range is 1 to 10.
- *Number of significant digits*- Specifies the number of significant digits to be printed. The default is 4 digits, and the allowed range is 2 to 10.

### Displaying Output Information

To display output information, choose *Info – Show Output*.


© 2023 Cadence Design Systems, Inc.

Output names are truncated to fit into columns if they are too long. The *Show Output* command shows the output names in full.

### Specifying Results to Print

Before you can print results, you need to specify which results to print.

1. Do one of the following:
   - Run a simulation.
   - In the ADE Explorer window, choose *Results – Select*, choose the desired data file, and click *OK*.
2. Make sure the Schematic window for the selected design is open.

To print results for the current simulation or for a selected data file,

1. Choose a print command from the *Results* menu.
2. Select a node in the Schematic window.

   The Results Display Window shows
   - The command syntax for the print option you selected
   - The results for the instance you selected

   Each time you click a node in the Schematic window, information about the node is added to the Results Display Window.

# Printing Components

Refer to the below topics:

- Printing DC Operating Points on page 818
- Printing Transient Operating Points on page 818
- Printing Model Parameters of Components on page 819
- Printing Noise Parameters of Nodes or Components on page 819
- Printing DC Mismatch Summary on page 831
- Printing AC Match Summary on page 832
- Printing Stability Summary on page 834


© 2023 Cadence Design Systems, Inc.

- Printing Pole Zero Summary on page 836
- Printing DC Node Voltages on page 838
- Printing Transient Voltages on page 839
- Printing Sensitivities on page 839

## Printing DC Operating Points

To print the DC operating points of the components in your circuit,

1. Choose *Results – Print – DC Operating Points*.
2. Move your cursor into the Schematic window.

   The CIW prompts you to select instances for the operating point output.
3. Click an instance.

If the selected instance is a textual subcircuit, operating points for all devices in the subcircuit will be printed. It may take some time to search for all instances in a textual subcircuit. To disable the feature, set the following environment variable in your `.cdsenv`:

```
asimenv.printing  printInlines  boolean  nil
```

## Printing Transient Operating Points

To print the final transient operating points of the nodes or components in your circuit,

1. Choose *Results – Print – Transient Operating Points*.
2. Move your cursor into the Schematic window.

   The CIW prompts you to select instances for the transient operating point (OPT) output.
3. Click an instance or node.

If the selected instance is a textual subcircuit, operating points for all devices in the subcircuit will be printed. It may take some time to search for all the instances in a textual subcircuit.To disable the feature, set the following environment variable in your `.cdsenv`:

```
asimenv.printing  printInlines  boolean  nil
```


© 2023 Cadence Design Systems, Inc.

## Printing Model Parameters of Components

To print the model parameters of the nodes or components in your circuit,

1. Choose *Results – Print – Model Parameters*.
2. Move your cursor into the Schematic window.

   The CIW prompts you to select instances for the model parameter output.
3. Click an instance of a device.

If the selected instance is a textual subcircuit, model parameter for all devices in the subcircuit will be printed. It may take some time to search for all instance in a textual subcircuit. To disable the feature, set the following environment variable in your `.cdsenv`:

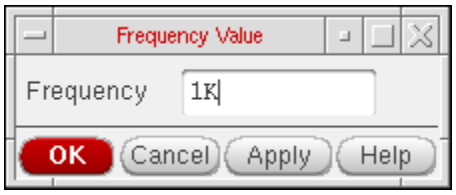
```
asimenv.printing  printInlines  boolean  nil
```

## Printing Noise Parameters of Nodes or Components

To print the noise parameters of the nodes or components in your circuit,

1. Choose *Results – Print – Noise Parameters*.

   The Select Frequency Value form appears.

   If the form does not appear, press `F3`.
2. In the *Frequency* field, type the frequency value at which you want the noise parameters to print.

   The default frequency is 1K.
3. Move your cursor into the schematic window.

   The CIW prompts you to select instances for the VNP output.
4. Click an instance or node.


© 2023 Cadence Design Systems, Inc.

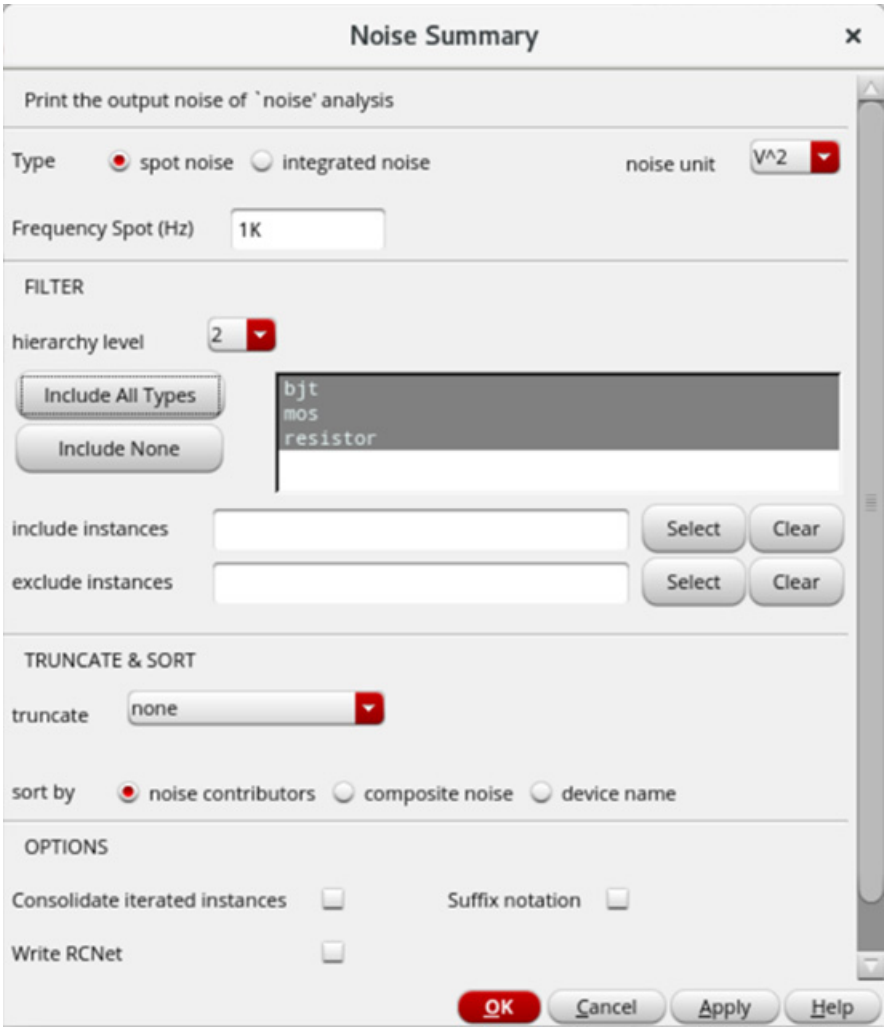
## Noise Summary

To display the noise contribution of the components in a circuit,

1. Run a noise analysis simulation.
2. Choose *Results – Print – Noise Summary*.

   Alternatively, right-click any cell in the *Test* or *Output* column of the *Results* tab and choose *Print – Noise Summary*.

   The Noise Summary form is displayed.




3. 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.


© 2023 Cadence Design Systems, Inc.

4. In the *FILTER* group box, do the following:

   a. From the *hierarchy level* drop-down list, specify the hierarchy levels up to which you want to view the noise contribution results in the *Results Display Window*.

   b. Specify the options to include or exclude device types or instances. Refer to the section Specifying Device Types and Instances to Include or Exclude below.

5. If you choose *integrated noise*, you have the option of using a weighting factor.

   The *flat* weighting factor specifies that the integration be performed on the original unweighted waveform.

   The *from weight file* selection specifies that, before the integration is performed, the noise contributions of particular frequencies in the original waveform be weighted by factors supplied from an input file. The weighting file must have one of the following entries on 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
```

6. Choose filtering details to include or exclude particular instances in your summary.

7. If needed, choose truncation details to shorten your summary. In the *TRUNCATE & SORT* group box, specify options to truncate and sort your noise summary data.

8. From the PARAMETRIC VARIABLES list, select a value of the design variable.

   **Note:** This list is displayed only for parametric dataset. It is recommended that you select the leaf dataset that you want to view the output for to launch the *Noise Summary* form.

9. Select the *Write RCNet* check box to print the information about parasitic resistors and their values for each physical wire net in the noise summary results. This provides users the information to diagnose the noise contribution issues related to abnormal wire nets.

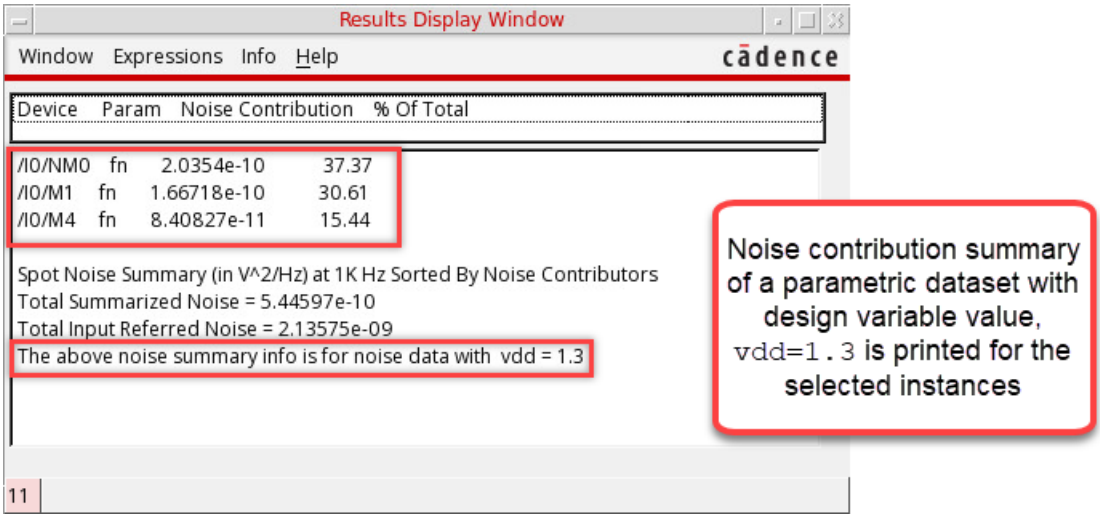
   This option is available only when,

   - *Spectre X* is selected as the simulation performance mode and *Preset* is set to *MX* in the High-Performance Simulation Options form.
   - The following statements are set in Spectre: `rcr_net_wire_noise = yes` and `rcr_net_res_info = yes`.

10. Click *OK*.


© 2023 Cadence Design Systems, Inc.

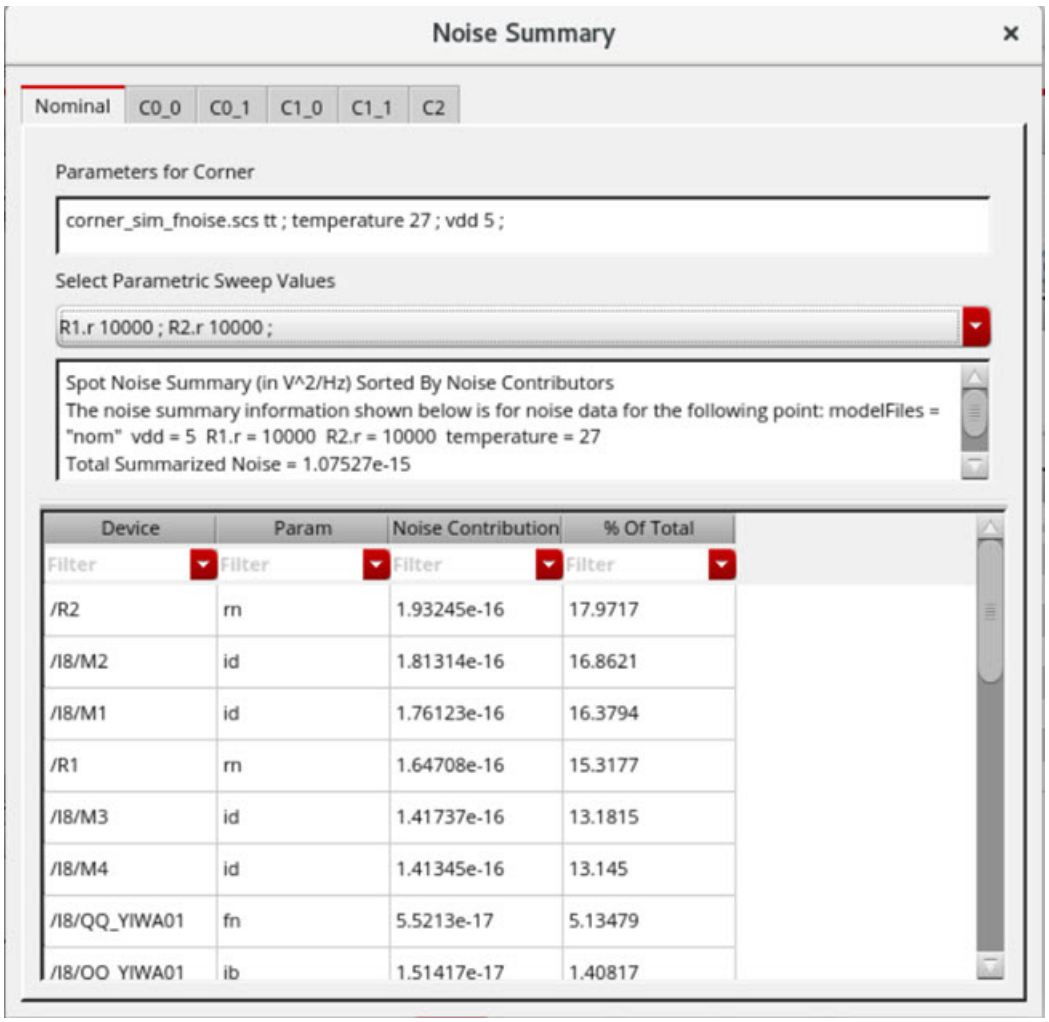
The Results Display window is displayed.




**Note:** Set the `maestro.gui showNoiseSummaryinTableWidget` environment variable to `t` in the `.cdsinit` file to see the Noise Summary window instead of the Results Display window. The Noise Summary window displays corner-specific noise


© 2023 Cadence Design Systems, Inc.

summary report in tabulated format.




You can shorten your summary by specifying the number of highest contributors to be included in the report, by specifying the percentage of noise a device must contribute to be included in the summary, or by specifying the level of noise a device must contribute to be included in the summary.

The precision of the noise data displayed in the Noise Summary window can be controlled using the environment variable `asimenv.noiseSummary digits`. The default value for this variable is 6 and can be set to any other integer value.

For example, the following command sets the value of the variable to `10`:

```
envSetVal("asimenv.noiseSummary" "digits" 'int 10)
```

The number of decimals printed for any relative contribution is controlled using the environment variable `asimenv.noiseSummary percentDecimals`. The default value for this variable is `2` and can be set to any other integer value.


© 2023 Cadence Design Systems, Inc.

For example, the following command sets the value of the variable to `4`:

```
envSetVal("asimenv.noiseSummary" "percentDecimals" 'int 4)
```

### Noise Summary Window

The Noise Summary window displays the noise summary report and has separate tabs for each corner.

**Note:** This window is displayed only when the variable `maestro.gui showNoiseSummaryinTableWidget` is set to `t`.

Each tab displays information for the following fields:

| Field | Description |
|---|---|
| *Parameters for Corner* | Displays the semicolon-separated list of parameters for a corner. |
| *Select Parametric Sweep Values* | Lets you select the parametric sweep values for which the results are to be displayed in the Noise Summary table.<br>According to the selected value, the information displayed in the text field below this drop-down list changes.<br>This text field displays the following information:<br>■ The noise summary type along with the sorting method<br>■ Points for which the noise data is calculated<br>■ Total summarized noise<br>■ Total input referred noise<br>■ Total shot noise |


© 2023 Cadence Design Systems, Inc.

| Field | Description |
|---|---|
| *Noise Summary table* | Displays the noise summary results as customized through the Noise Summary form.<br><br>This table has the following columns:<br><br>■ *Device*: Specifies the name of the device.<br><br>■ *Param*: Specifies the name of the parameter.<br><br>■ *Category*: Specifies the noise category.<br><br>**Note:** To see this column, set the SHELL variable `SPECTRE_NOISETYPE=1`.<br><br>■ *Noise Contribution*: Displays the noise contribution for the given device.<br><br>■ *% Of Total*: Displays the percentage of noise that the given device has contributed to the total summarized noise. |

***Related Topics***

maestro.gui showNoiseSummaryinTableWidget

### 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.


© 2023 Cadence Design Systems, Inc.

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.

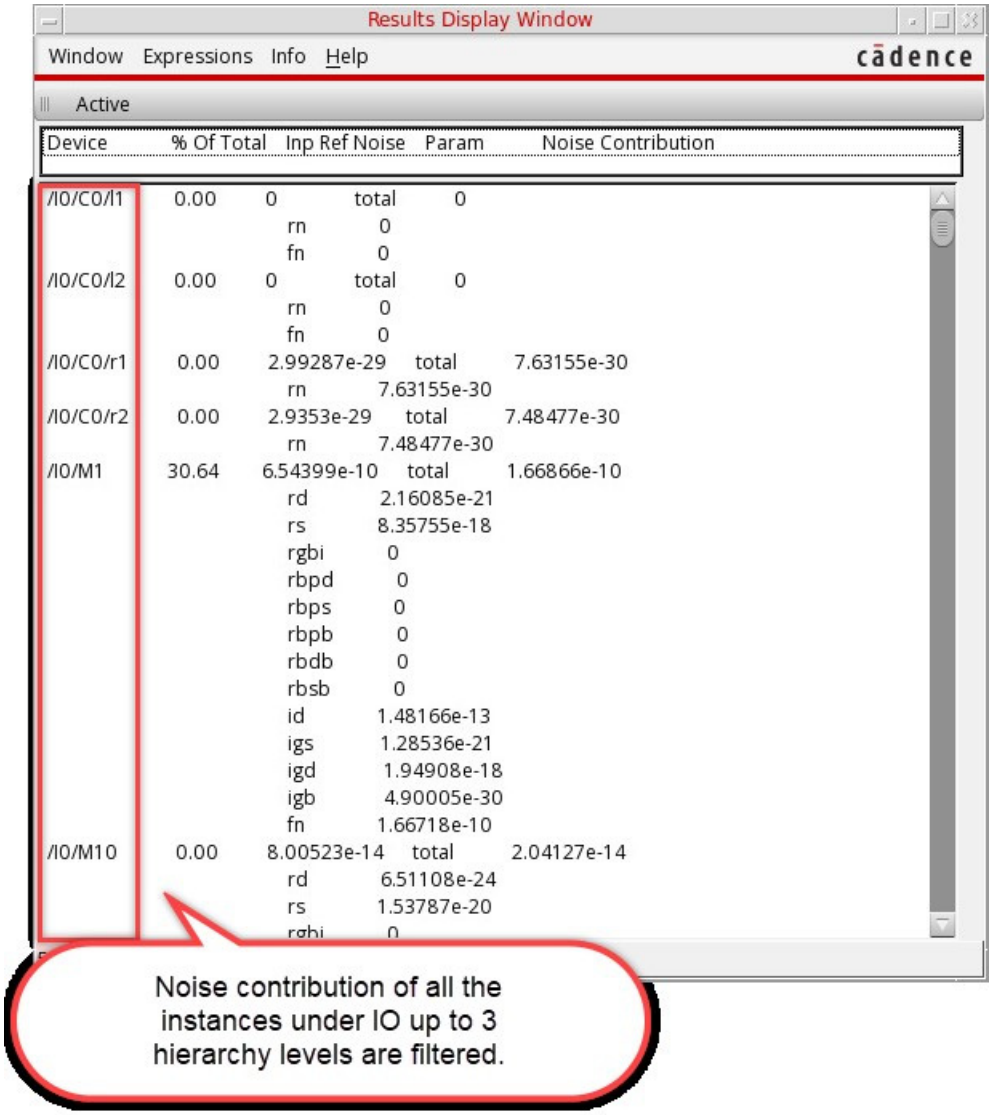
   Each instance path appears in the *include instances* field.
3. Press *Esc* when you are finished selecting.

Alternatively, in the include instances field, you can type the instance names for which you want to filter the noise contribution results. For example, if you want to view the noise contribution only for the `/I0/C0` instance, type `/I0/C0`.

**Note:** You can also use the wildcard character `*` to filter the results. For example, if you want to view the noise contribution for all the instances under `I0`, type `/I0/*` in the *include*


© 2023 Cadence Design Systems, Inc.

*instances* field. The Results Display window shows the results for all the instances under `IO` up to the specified hierarchy levels, as shown in the following figure.




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.


© 2023 Cadence Design Systems, Inc.

### 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*.


© 2023 Cadence Design Systems, Inc.

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.

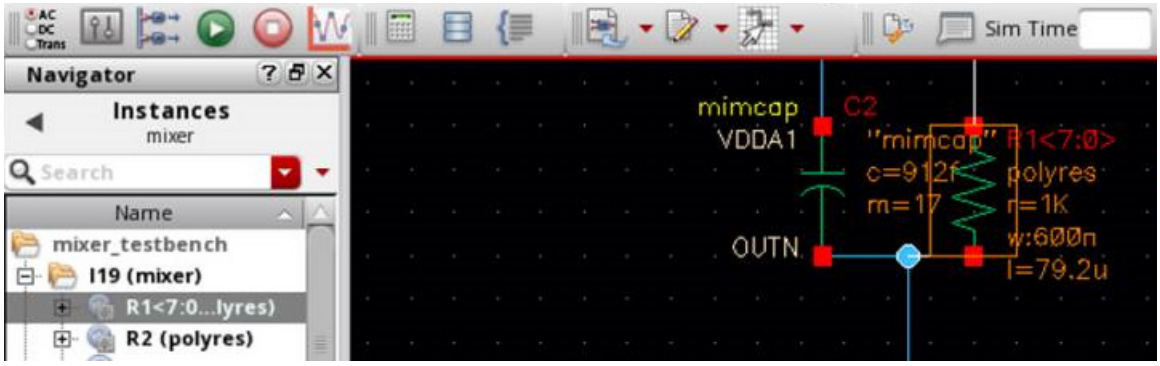
You can select one, two, or three check boxes. The number of boxes you select determines the number of lists that appear in your Results Display Window when you click *Apply* or *OK*.

### Specifying the options to customize the printed Noise Summary data

The *OPTIONS* group provides the following options to customize the way noise summary is printed:

- *Consolidate iterated instances* – Combines the noise contributions of iterated instances to one contributor.

  For example, consider an iterated instance `R1`




  If the consolidate iterated instances feature is turned off, the noise contributions of the iterated instances are not combined.

  When this feature is turned on, you can see the combined noise contributions of the iterated instances.

- *Suffix notation* – Changes the notation of the noise contribution value for the data output to the specified value.

  For example, `3.36916e-05`, is shown as `33.6916u`.

  Default value: `nil`


© 2023 Cadence Design Systems, Inc.

### Controlling Precision of Printed Noise Data

You can control the precision of the 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
```


© 2023 Cadence Design Systems, Inc.

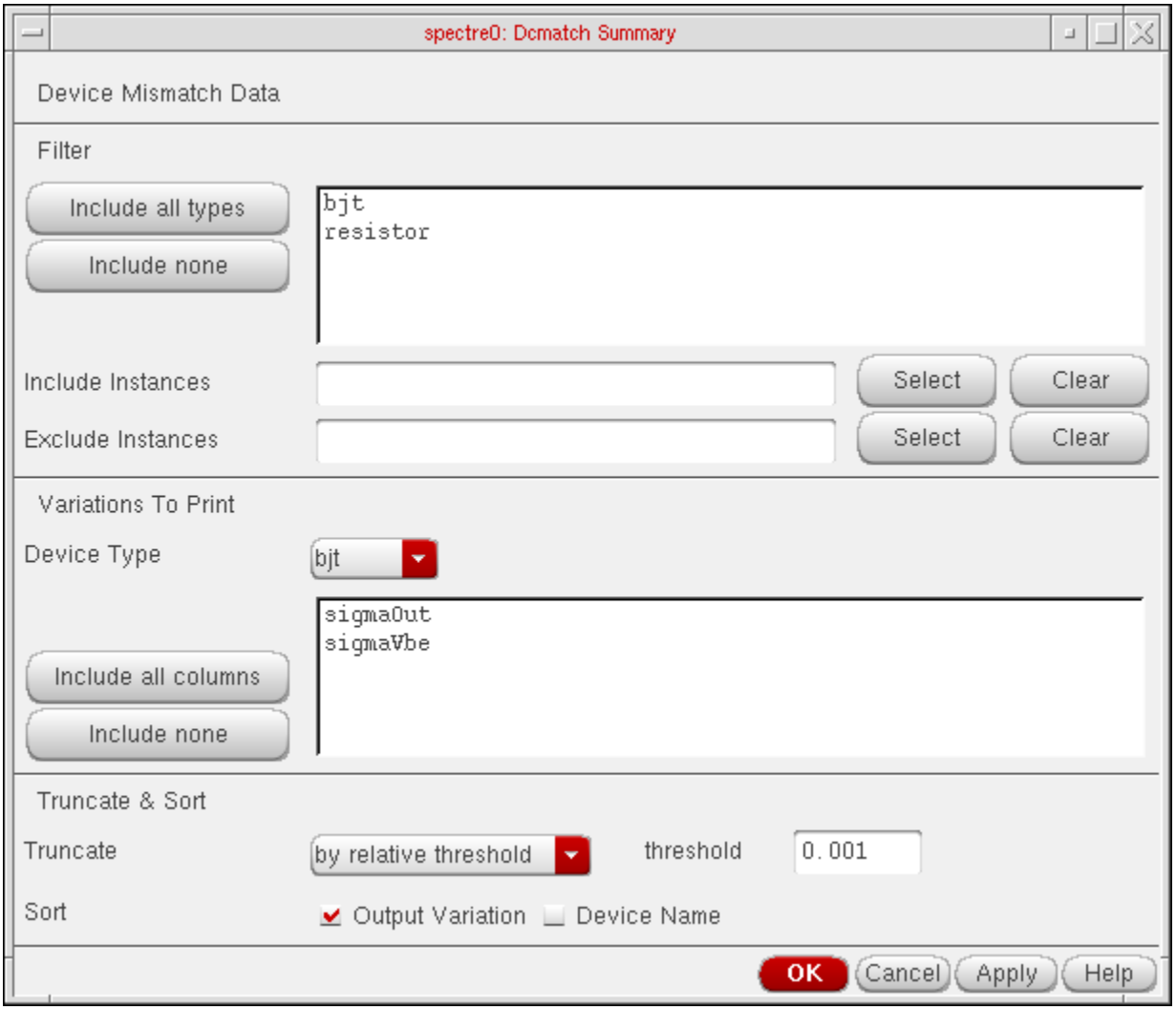
## Printing DC Mismatch Summary

To print the DC Mismatch summary in your circuit,

1. Choose *Results – Print – Mismatch Summary.*

   **Note:** This menu option is enabled when ever analysis is included in the last run or the results directory specifically selected through ADE Explorer, contains the results for analysis.

   The *DC Mismatch Summary* form appears




2. Specify a value in the *Print results when value is* field.


© 2023 Cadence Design Systems, Inc.

3. Specify the type of devices you need to print the results for, in the *Filter* section. The *Include all types* and *Include none* buttons can be used to include or exclude all types at a single click. You can include specific instances or exclude specific instances. You can either type the instance names or use the select buttons to pick them from schematics. The *Clear* button is used to clear the fields.
4. Specify the information to be made available for the various device types, in the *Variations to Print* section. The *Include all columns* and *Include none* buttons can be used for easier list box operation.
5. Truncate and sort data by top contributors and relative/absolute contribution. The default is relative contribution with the threshold being the value of the threshold parameter used on the analysis line. You can sort by variation or device name.

## Printing AC Match Summary

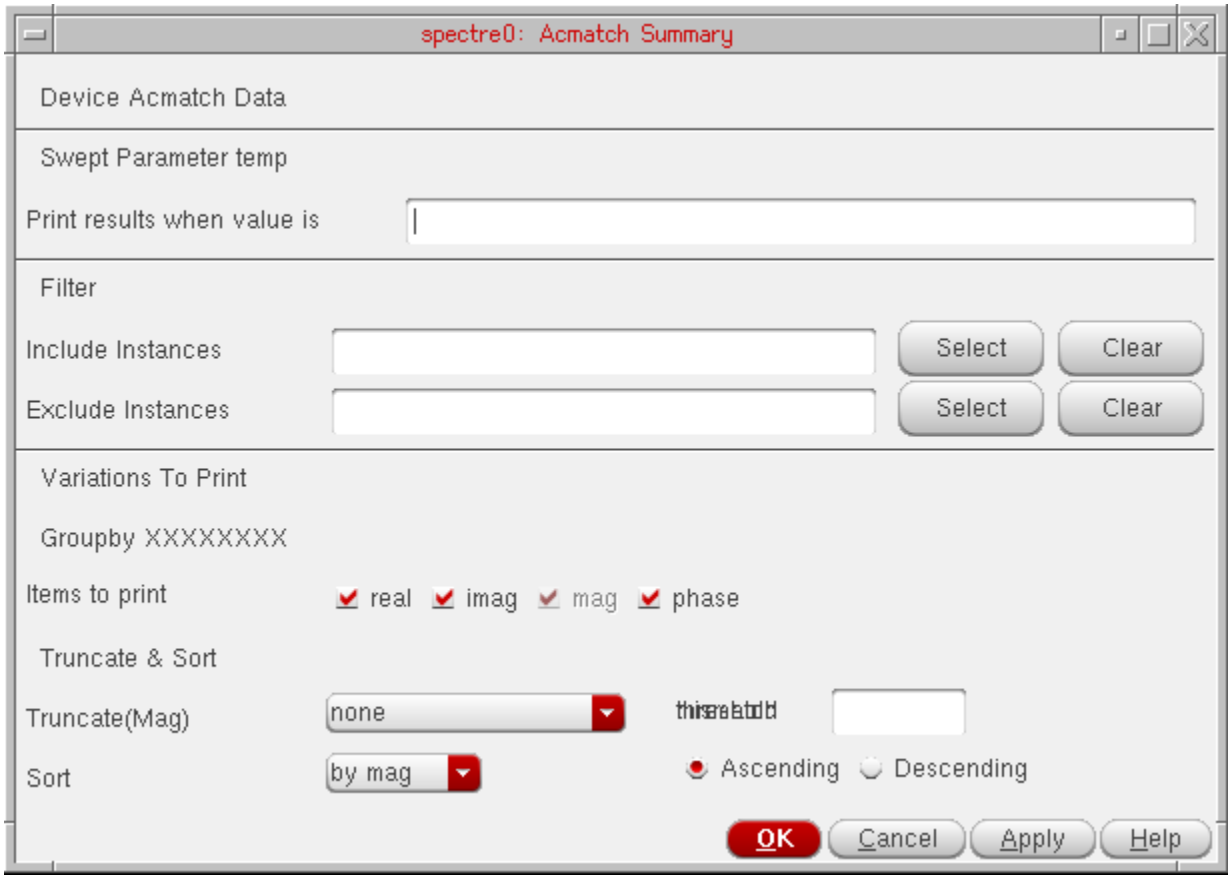
To print the AC match summary in your circuit,

- Choose *Results – Print – ACmatch Summary.*

  **Note:** This menu option is enabled only when acmatch analysis is included in the last run or the results directory specifically selected through ADE Explorer, contains the results for acmatch analysis.


© 2023 Cadence Design Systems, Inc.

The *AC Mismatch Summary* form appears, as shown in the figure below.



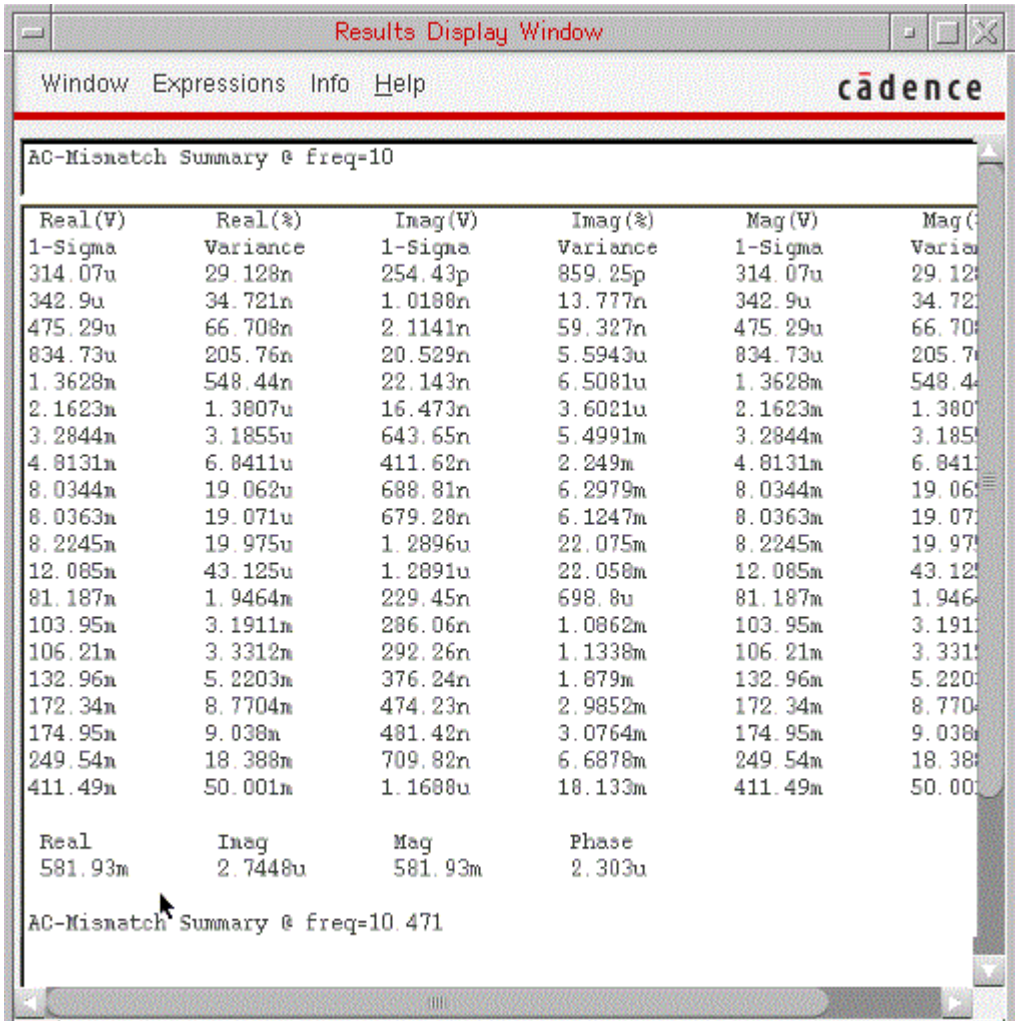

In this form, specify the following fields:

- In the *Swept Parameter Temp* section, specify the value for temperature in the *Print results when value is* field.
- In the *Filter* section, specify the instances that you want to include or exclude. You can either type the instance names or use the *Select* buttons to select them from schematic. The *Clear* button is used to clear the fields.
- In the *Variations to Print* section, select the items you want to print—`real`, `imag`, `mag`, and `phase`.
- In the *Truncate & Sort* drop-down list, specify how you want to truncate and sort data. Available options to truncate the data are none, by relative threshold, or by absolute threshold. Default value is none. You can also specify the threshold value to be used in the Threshold field.


© 2023 Cadence Design Systems, Inc.

- In the *Sort* drop-down, specify how you want to sort the data. Available options are `by mag` or `by name`. Default value is by mag. Select either of the *Ascending* or *Descending* options to specify which sorting order you want to apply.

The *Results Display Window* displays the acmatch analysis run summary using the criteria you specified.




## Printing Stability Summary

To print the stability summary in your circuit,

1. Choose *Results – Print – Stability Summary.* The *Stability Summary* form appears.The form enables you to print *Phase Margin*, *Gain Margin* or *Both*.

   **Note:** This menu option will be enabled only when stability analysis was included in the


© 2023 Cadence Design Systems, Inc.

last run or the results file exists in the results directory when you specifically selected an existing results directory through ADE Explorer.

The form handles parametric (family) data and prints results at all available sweep points.

2. Choose the required data and click *OK*.

   The *Results Display Window* displays the stability summary using the criteria that you specified. For example, if you had swept temperature and capacitor values with the


© 2023 Cadence Design Systems, Inc.

parametric tool for the stability analysis and selected the *Both* option on the form, the *Results Display Window* will appear as follows:

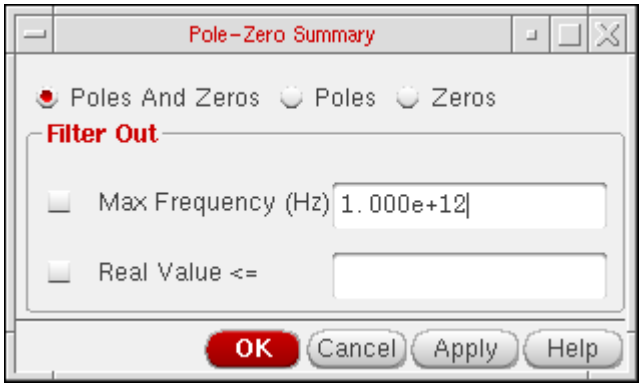
## Printing Pole Zero Summary

To print the *Pole Zero* summary in your circuit,


© 2023 Cadence Design Systems, Inc.

1. Choose *Results – Print – Pole Zero Summary.* The *Pole-Zero Summary* form appears. The form enables you to print poles or zeros, or poles and zeros with filtering options.

   **Note:** This menu option will be enabled only when pole zero analysis was included in the last run or the results file exists in the results directory when you specifically selected an existing results directory through ADE Explorer.




2. Select the option, *Poles* if you want to plot only poles, *Zeros* if you want to plot only zeros and *Poles and Zeros* if you want to plot both poles and zeros.

   **Note:** By default, the option *Poles and Zeros* is selected.

3. Set the required options in the *Filter Out* section and click *OK*. This section provides a combination of filtering mechanisms that you can select in order to plot the poles and zeros. These are:

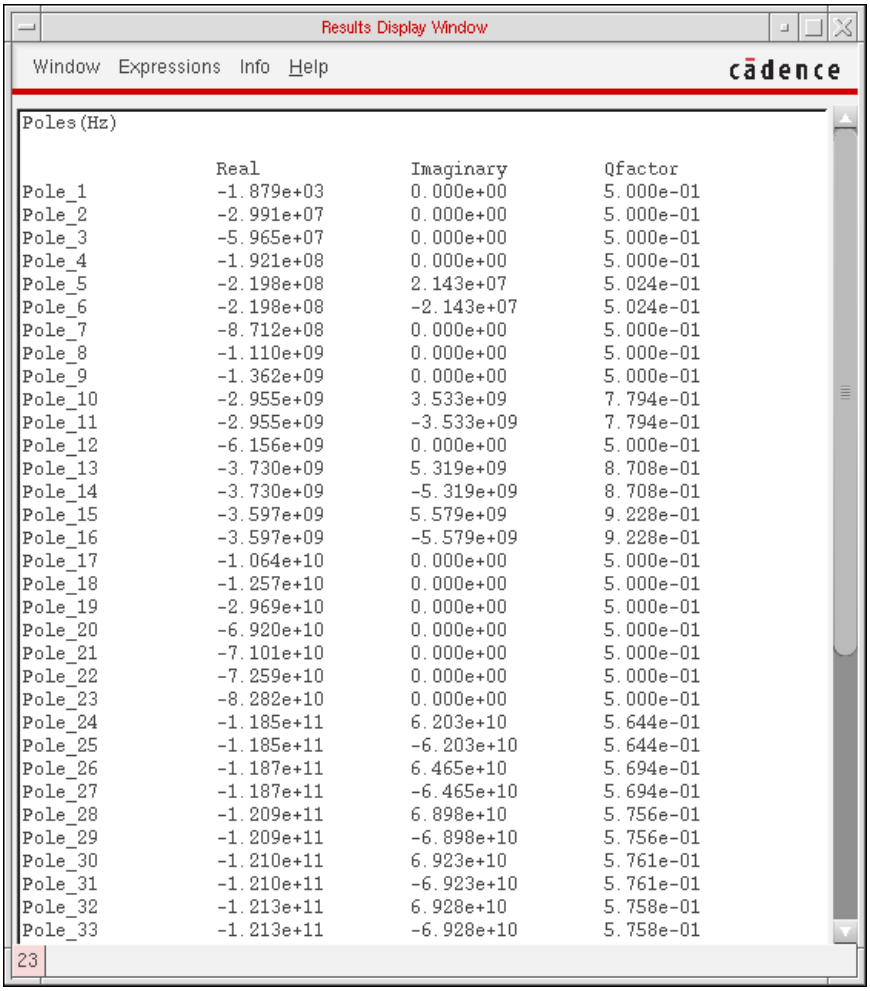
   - *Max Frequency:*
     This option enables you to filter out poles and zeros that are outside the frequency band of interest (FBOI) and that do not influence the transfer function in the FBOI. The default value is that specified in the *fmax* field in the *Pole-Zero Options* form. Note, that for the *Direct Plot* form, *fmax* is read from the header of the `psf` data. Only poles and zeros whose magnitudes exceed the frequency value specified are filtered out.

   - *Real Value:*
     This option enables you to specify the real part of the frequency. Only poles and zeros whose real values are less than or equal to the real value specified are filtered out.

   **Note:** By default, no filtering is selected. You can set the filtering criteria once you specify either poles or zeros or both to be plotted.


© 2023 Cadence Design Systems, Inc.

4. The *Results Display Window* displays the pole zero summary using the criteria that you specified:




## Printing DC Node Voltages

To print the DC node voltages of the nodes or components in your circuit,

1. Choose *Results – Print – DC Node Voltages*.
2. Move your cursor into the Schematic window.

   You are prompted to select nets for the VDC output.
3. Click a node.


© 2023 Cadence Design Systems, Inc.

## Printing Transient Voltages

To print the transient node voltages of the nodes in your circuit,

1. Choose *Results – Print – Transient Node Voltages*.

   The Schematic window comes into the view and the *Select Time Value* form appears.

2. In the *Time* field, type the time value at which you want to print the transient node voltages. The default time value is `0`.
3. Select nets, for which you want to view the voltage value, on the schematic.

   For each selected net, the voltage value is displayed in the *Results Display Window*.
4. Press <Esc> after the selection is complete.

## Printing Sensitivities

To print the sensitivities in your circuit,

1. Choose *Results – Print – Sensitivities*.
2. Move your cursor into the Schematic window.

   You are prompted to select nets for the output.
3. Click a net or port.

## Printing Capacitance Data

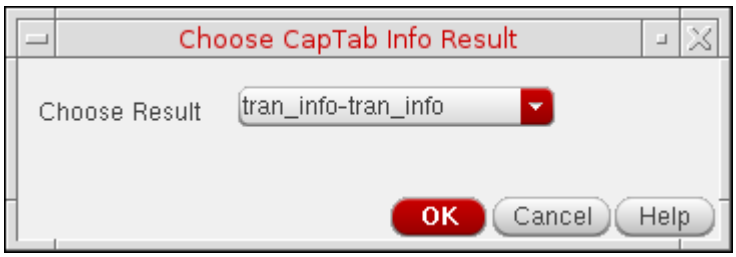
When you run a transient or dc analysis with the *captab* option selected in the analysis option.

To view capacitance data:

1. Choose *Results - Print - Capacitance Table*.

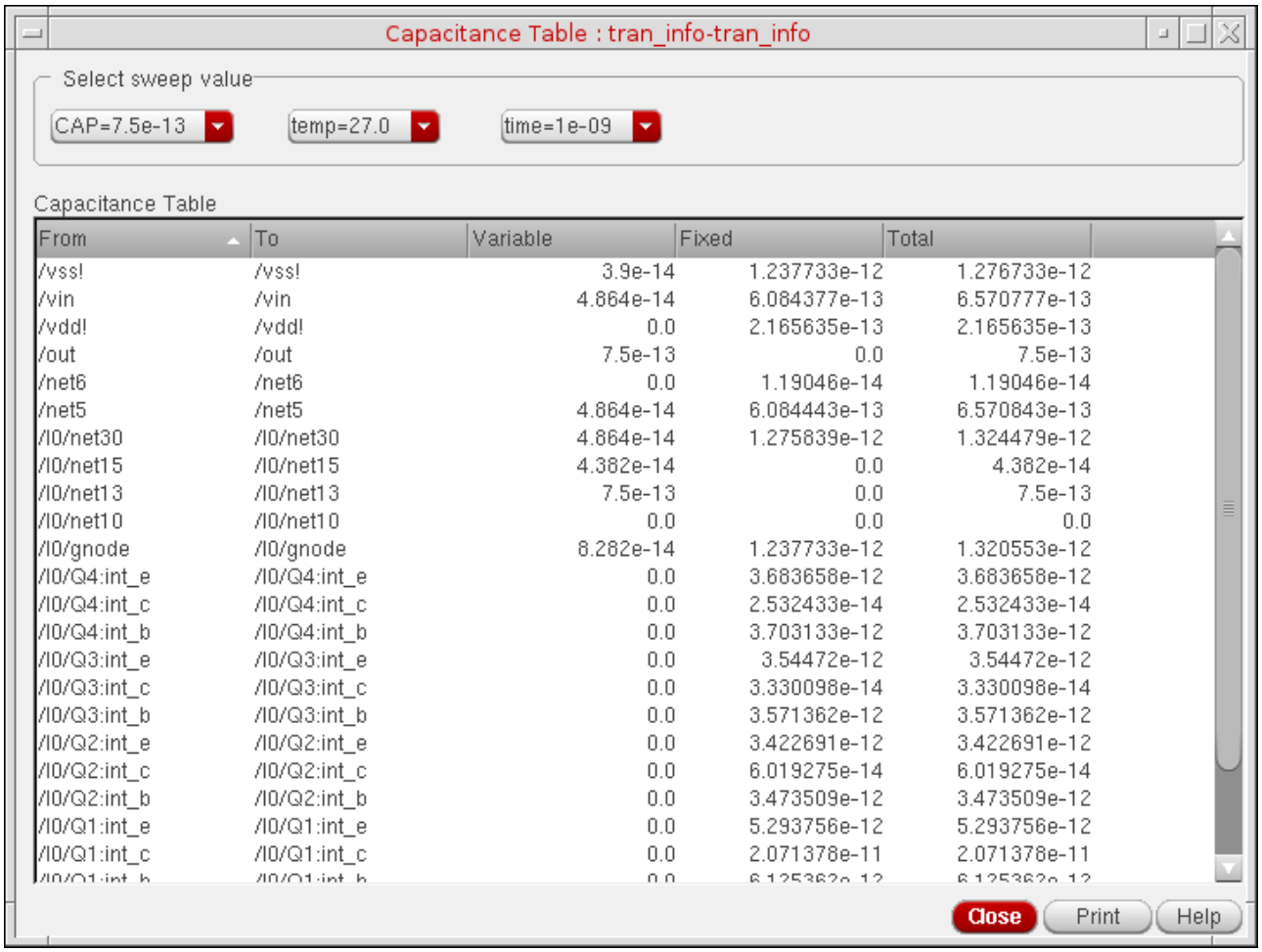

© 2023 Cadence Design Systems, Inc.

If you had opted to save captab data for more than one analysis, the *Choose CapTab Info Result* form appears first.




Select one captab result file from the *Choose Result* cyclic box and click *OK*.

The Capacitance Table appears as shown below:




| From | To | Variable | Fixed | Total |
|---|---|---|---|---|
| /vss! | /vss! | 3.9e-14 | 1.237733e-12 | 1.276733e-12 |
| /vin | /vin | 4.864e-14 | 6.084377e-13 | 6.570777e-13 |
| /vdd! | /vdd! | 0.0 | 2.165635e-13 | 2.165635e-13 |
| /out | /out | 7.5e-13 | 0.0 | 7.5e-13 |
| /net6 | /net6 | 0.0 | 1.19046e-14 | 1.19046e-14 |
| /net5 | /net5 | 4.864e-14 | 6.084443e-13 | 6.570843e-13 |
| /I0/net30 | /I0/net30 | 4.864e-14 | 1.275839e-12 | 1.324479e-12 |
| /I0/net15 | /I0/net15 | 4.382e-14 | 0.0 | 4.382e-14 |
| /I0/net13 | /I0/net13 | 7.5e-13 | 0.0 | 7.5e-13 |
| /I0/net10 | /I0/net10 | 0.0 | 0.0 | 0.0 |
| /I0/gnode | /I0/gnode | 8.282e-14 | 1.237733e-12 | 1.320553e-12 |
| /I0/Q4:int_e | /I0/Q4:int_e | 0.0 | 3.683658e-12 | 3.683658e-12 |
| /I0/Q4:int_c | /I0/Q4:int_c | 0.0 | 2.532433e-14 | 2.532433e-14 |
| /I0/Q4:int_b | /I0/Q4:int_b | 0.0 | 3.703133e-12 | 3.703133e-12 |
| /I0/Q3:int_e | /I0/Q3:int_e | 0.0 | 3.54472e-12 | 3.54472e-12 |
| /I0/Q3:int_c | /I0/Q3:int_c | 0.0 | 3.330098e-14 | 3.330098e-14 |
| /I0/Q3:int_b | /I0/Q3:int_b | 0.0 | 3.571362e-12 | 3.571362e-12 |
| /I0/Q2:int_e | /I0/Q2:int_e | 0.0 | 3.422691e-12 | 3.422691e-12 |
| /I0/Q2:int_c | /I0/Q2:int_c | 0.0 | 6.019275e-14 | 6.019275e-14 |
| /I0/Q2:int_b | /I0/Q2:int_b | 0.0 | 3.473509e-12 | 3.473509e-12 |
| /I0/Q1:int_e | /I0/Q1:int_e | 0.0 | 5.293756e-12 | 5.293756e-12 |
| /I0/Q1:int_c | /I0/Q1:int_c | 0.0 | 2.071378e-11 | 2.071378e-11 |

The table shows captab data in five columns:

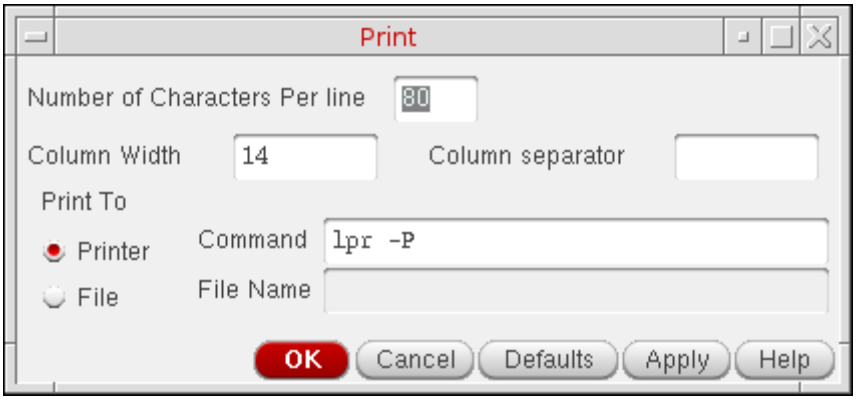
- *From*: The nodes from which capacitance is measured.


© 2023 Cadence Design Systems, Inc.

- *To*: The nodes to which capacitance is measured.
- *Variable*: Variable capacitance between nodes.
- *Fixed*: Fixed capacitance between nodes.
- *Total*: Total capacitance between nodes.

2. Select one of more sweep values for which the capacitance table is to be printed by using the cyclic buttons in the Select sweep value group box. The values in the *Capacitance Table* change dynamically as you select the different options.
3. Click the *Print* button to save or print captab data.

   The Print form appears.




4. Specify values in this form as follows:
   a. *Number of characters per Line*: Number of characters to be printed on a line.
   b. *Column Width*: Width of the columns to be printed. The default value is 14.
   c. *Column Separator*: Column separator to be used for printing the table. The default is a space.
   d. *Print To*: If you want to print to a printer, select the *Printer* option button and specify a command. If you want to print to a file, select the *File* option button and specify a filename.
5. Click *OK* or *Apply*.

## Precision Control for Printing

Precision of printed results can be controlled using `aelPushSignifDigits`.


© 2023 Cadence Design Systems, Inc.

### 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
```

# 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")` |


© 2023 Cadence Design Systems, Inc.

| To display… | Type this command in the CIW |
|---|---|
| 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 data onto the schematic to show the parameters, operating points, net names, currents and voltages of the design components. You can also change the existing annotations using the context menu or the annotation setup form.

To annotate data on the schematic,

➤ In the ADE Explorer window, choose *Results* and then one of the annotation commands.

To annotate instances selectively, use the *View – Annotations – Setup* command in the Schematic window. For more information on annotating data on the Schematic, see Annotating Data on the Schematic Window section in the *Virtuoso Schematic Editor L User Guide*.


- Annotating DC Node Voltages on page 844
- Annotating DC Operating Points on page 844
- Annotating DC Currents on page 844
- Annotating Transient Voltages on page 844
- Annotating Transient Currents on page 845
- Annotating Transient Operating Points on page 845
- Specifying the Data Directory for Labels on page 846
- Saving and Removing Annotated Labels on page 848



© 2023 Cadence Design Systems, Inc.

## Annotating DC Node Voltages

To annotate DC node voltages,

➡ In the Simulation window, choose *Results – Annotate – DC Node Voltages*.

Alternatively, right-click in the *Results* tab and choose *Annotate – DC Node Voltages.*

The annotations for DC node voltages that provide information about the node voltages are displayed on the schematic.

## Annotating DC Operating Points

To annotate DC operating points,

➡ In the Simulation window, choose *Results – Annotate – DC Operating Points*.

Alternatively, right-click in the *Results* tab and choose *Annotate – DC Operating Points.*

The annotations for DC operating points that provide information about the operating point parameters are displayed on the schematic.

## Annotating DC Currents

To annotate DC currents,

➡ In the Simulation window, choose *Results – Annotate – DC Currents*.

Alternatively, right-click in the *Results* tab and choose *Annotate – DC Currents.*

The annotations for DC currents that provide information about the currents are displayed on the schematic.

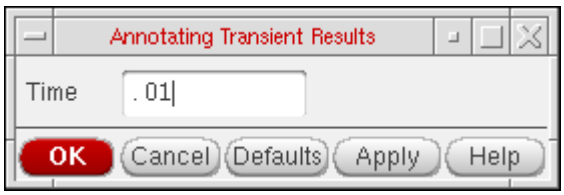
## Annotating Transient Voltages

To annotate transient voltages,

1. In the Schematic window, choose *View – Annotations – Transient Node Voltages*.


© 2023 Cadence Design Systems, Inc.

The Annotating Transient Results form appears.




2. Type the transient time point in the *Time* field, and click *OK*.

   Alternatively, you can also type the transient time points in the *Sim Time* text box on the Schematic toolbar.

## Annotating Transient Currents

To annotate transient currents:

1. In the Schematic window, choose *Results – Annotate – Transient Currents*. Alternatively, right-click in the Results tab and choose Annotate - Transient Currents

   The Annotating Transient Results form appears.

2. Type the transient time point in the *Time* field, and click *OK*.

   Alternatively, you can also type the transient time points in the *Sim Time* text box on the Schematic toolbar.

## Annotating Transient Operating Points

To annotate final transient operating points,

1. In the Simulation window or the Schematic window, choose *Results – Annotate – Transient Operating Points*. This will annotate the operating point data for the final timepoints.

To annotate infotimes transient operating points,

1. In the Simulation window or the Schematic window, choose *Results – Annotate – Transient Operating Points*.

   The *Annotating Transient Operating Points* form appears.

2. Select the transient time point in the *Time* drop-down field. This field lists the choices of timepoints at which the operating point data is stored.This will annotate the operating point data for the selected timepoint saved.


© 2023 Cadence Design Systems, Inc.

3. Click *OK* or *Apply*. These two buttons essentially perform the same operation except that the Apply button does not close the form enabling the user to select another timepoint and click *Apply* again to annotate data for a different timepoint. Clicking on the *Cancel* button will cancel entire operation.

**Important Points to Note:**

- The options to annotate are enabled depending on the availability of the results. For example, if you do not have the results for transient simulation, the options to plot and annotate the transient voltages, transient currents, and transient operating points will be disabled.
- This form will not come up if the user has not stored operating point data at different timepoints.
- If the user wants to save the operating point data, in the Transient Options form, choose Output and fill the timepoints in the field infotimes.

  For more information, see Transient Analysis in the *Spectre Circuit Simulator Reference*.
- *The Results – Annotate – Show Parasitics* and *Results – Annotate – Hide Parasitics* are enabled only when the results are available for *dcop*.
- To annotate transient operating points for HSpiceD simulator, you must enable the OP Analysis.

## Annotating AC Operating Points

To annotate AC operating points,

➥ Choose *Results – Annotate – AC Operating Points*.

The annotations for AC operating points are dispalyed on schematic and provides the information about operating point parameters.

## Specifying the Data Directory for Labels

To specify the simulation data directory (run directory) for labels,

1. In the Schematic window, choose *View – Annotations – Setup.*


© 2023 Cadence Design Systems, Inc.

The Annotation Setup form appears.

2. In the *Simulation Data Directory* field, type the path to the simulation run directory and click *OK*.

**Note:** You do not need to use this form if

- You have specified the correct directory as the run directory


© 2023 Cadence Design Systems, Inc.

- You used the Results Browser to select results for the current schematic
- The most recent simulation you ran was of this schematic

## Saving and Removing Annotated Labels

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*.

## Conditional Search for Results

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, choose *Results – Circuit Conditions* from the *Simulation* menu. Follow the procedure below to search for circuit conditions.

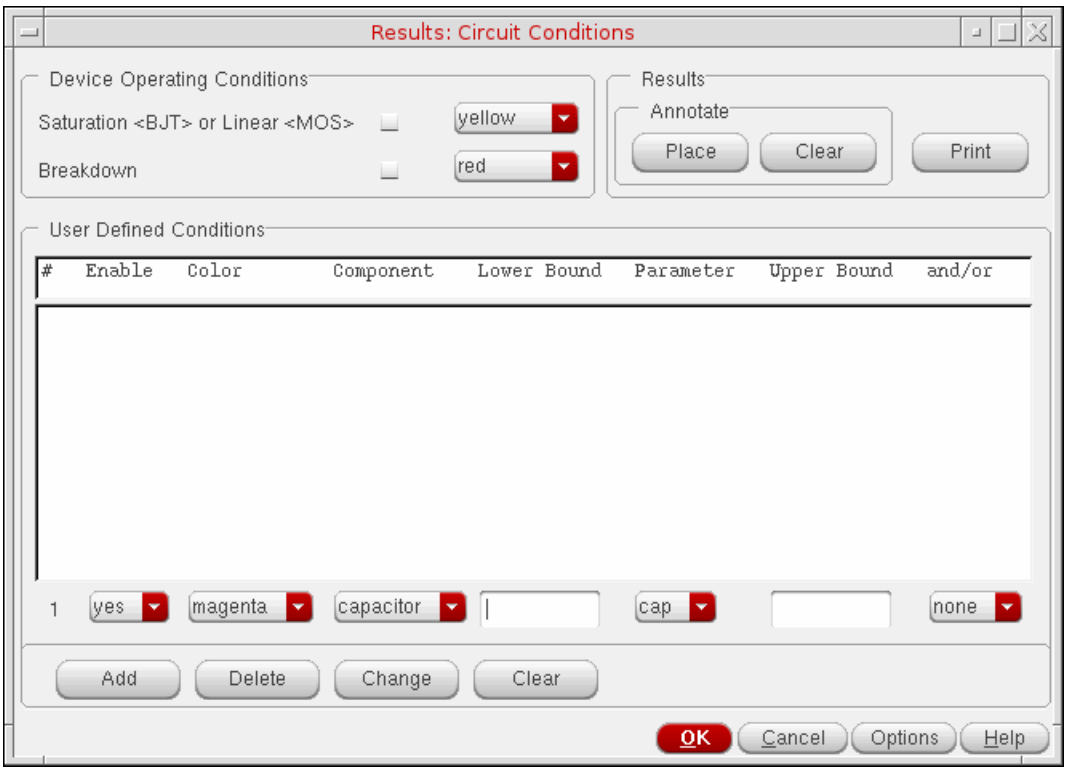
1. Run a simulation.

   **Note:** You must run a DC operating-point analysis to use the circuit conditions capability.

2. Choose *Results – Select* and indicate the results that you wish to search.
3. Choose *Results – Circuit Conditions* from the Simulation window.


© 2023 Cadence Design Systems, Inc.

The *Circuit Conditions* form appears:




For detailed information about the form, see Circuit Conditions on page 856.

4. Choose 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.

5. Set up *User Defined Conditions*.

   You use the cyclic and type-in fields to create the custom conditions you want to search for.

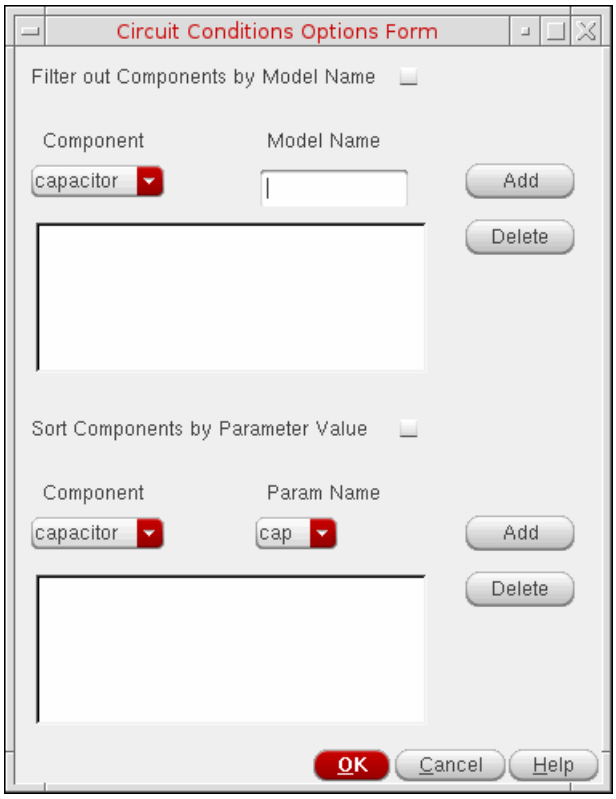
6. View the results of the conditions you chose by doing the following.

   - Click *Place* to highlight the instances that meet the specified conditions on the schematic.
   - Click *Print* to print the values of instances that meet the specified conditions in a print window.


© 2023 Cadence Design Systems, Inc.

- Click *Clear* to de-highlight all the instances on the schematic.

7. Clicking on the *Options* button will bring up a form where you can specify filter and sort conditions.




In the *Filter out Components by Model Name* section, you can enter filters using the cyclic field displaying all the component types and the text entry field to type in model names. After you have selected the component type and entered a model name, press *Add* to add the filter to list of filters. You can select one or more filters in the list and then click *Delete* to delete the filters. The filters are active only when the *Filter out Components by Model Name* check box is selected. When the filters are active, any component that matches a filter will be filtered out when you click the *Place* button or the *Print* button.

The next section is *Sort components by Parameter Value*. Users can use the two cyclic fields to enter sorting criteria for a component type. When this section is active (Boolean is on) the output from *Print* for user defined conditions will be sorted according to the sort variable for given component type.


© 2023 Cadence Design Systems, Inc.

# Form Field Descriptions

## Setting Plotting Options

**Print After** refers to the commands located in the *Plot Outputs* menu.

**Each Selection** specifies that the plot is printed after each node is selected.

**All Selections Are Made** specifies that none of the plots are printed until all of the nodes have been selected. You can select more than one node and click the `Escape` key when finished, and all the selected nodes are printed at the same time (into a table).

**Direct Plots Done After** refers to the commands located in the *Direct Plot* menu.

**Each Selection** specifies that the plot is drawn after each node is selected.

**All Selections Are Made** specifies that none of the plots are drawn until all of the nodes have been selected. You can select more than one node and click the `Escape` key when finished, and all the selected nodes are printed at the same time (into a table).

For the calculator `print` and `printvs` functions, you can use append mode and have more than one expression in the buffer and use `print` or `printvs` to print into a table.

**Annotations** selects information to be displayed in the graph window.

**Design Name** displays the design name in the graph window.

**Simulation Date** displays the simulation run date in the graph window.

**Temperature** displays the temperature associated with the plotted results in the graph window.

**Design Variables** displays the names and values of user-created variables in the graph window.

**Scalar Outputs** displays simulation results that evaluate to scalar values in the graph window.


© 2023 Cadence Design Systems, Inc.

## XF Results

### Plotting Mode

**Append** adds the new plot to existing plots that are already displayed in the graph window.

**Replace** replaces existing plots with the new plot.

**New SubWin** adds the plot to a new subwindow.

**New Win** adds the plot to a new window.

### Function

**Voltage Gain** is a calculation of voltage over voltage.

**Transimpedance** is a calculation of voltage over current.

**Current Gain** is a calculation of current over current.

**Transconductance** is a calculation of current over voltage.

### Modifier

**Magnitude** (the default setting) plots the magnitude of the selected signal.

**Phase** plots the phase of the selected signal.

**dB20** plots the magnitude in dB20.

**Real** plots the real component of the signal.

**Imaginary** plots the imaginary component of the signal.

**Replot** triggers the plotting of the selected instance or instance terminal with modified specifications.

**Add To Outputs** followed by **Replot** adds the output to the *Table Of Outputs* list box in the Simulation window.

**Select instance on schematic** or **Select instance terminal on schematic** prompts you to select the appropriate instance or terminal from the schematic.


© 2023 Cadence Design Systems, Inc.

## S-Parameter Results

### Plotting Mode

**Append** adds the new plot to existing plots that are already displayed in the graph window.

**Replace** replaces existing plots with the new plot.

**New SubWin** adds the plot to a new subwindow.

**New Win** adds the plot to a new window.

**Function** specifies the S-parameter or noise-parameter function to plot.

**SP** is S-parameters.

**ZP** is Z-parameters.

**YP** is Y-parameters.

**HP** is H-parameters.

**GD** is group delay.

**VSWR** is voltage standing wave ratio.

**NFmin** is minimum noise figure.

**Gmin** is the source reflection coefficient corresponding to NFmin.

**Rn** is equivalent noise resistance.

**NF** is noise figure.

**B1f** is the intermediate term for Kf, the Rollet stability factor.

**Kf** is the Rollet stability factor.

**GT** is transducer gain.

**GA** is available gain.

**GP** is power gain.

**NC** is noise circles.

**GAC** is available gain circles.

**GPC** is power gain circles.

**LSB** is load stability circles.


© 2023 Cadence Design Systems, Inc.

**SSB** is source stability circles.

**Plot Type** specifies the plot format. Option availability is a function of the selected function.

**Auto** uses the format in the current graph window unless that format is unsuitable for the function.

**Rectangular** specifies curves plotted against frequency.

**Z-Smith** specifies curves plotted on a Smith chart with impedance overlay.

**Y-Smith** specifies curves plotted on a Smith chart with admittance overlay.

**ZY-Smith** plots the signals to an Immittance Smith chart. In the Immittance Smith chart (ZY-Smith), both Z Smith and Y Smith grids are plotted on the same chart.

The Immittance Smith Chart has two grids: red and green. The red grid represents the Impedance or Z Smith grid, whereas the green grid represents the Admittance or Y Smith grid. On the red grid, the region above the x axis represents inductive impedances (positive imaginary parts) and the region below the x axis represents capacitive impedances (negative imaginary parts)

**Polar** specifies curves plotted in polar (mag/angle) coordinates.

**Modifier**, which is used only for rectangular plots, specifies the modifier the analog circuit design environment uses to reduce complex data for two-dimensional presentation. Option availability depends on the selected function; some functions, such as stability factor, do not require a modifier.

**Magnitude** plots the magnitude of complex or scalar quantities.

**Phase** plots the phase of complex quantities in degrees.

**dB20** plots the magnitude in dB.

**Real** plots the real part of complex quantities.

**Imaginary** plots the imaginary part of complex quantities.

**Sweep** selects a set of circles to be plotted against frequency or dB. (Sweep appears on the form only when you are plotting circles and have selected the NC, GAC, or GPC function.)

You can plot noise and gain circles at a single dB value for a range of frequencies or at a single frequency for a range of dB values.

When plotting stability circles, you can specify a frequency range. Use SSB to plot stability circles at the input port, and use LSB to plot those at the output port. You can specify a limited frequency range for these contours.


© 2023 Cadence Design Systems, Inc.

**Level (dB)** specifies the gain or noise figure value in dB for circles plotted against frequency.

**Frequency Range** defines *Start*, *Stop*, and *Step* for circles plotted at the specified dB value.

If you do not type in values for the frequency range, a circle is plotted for every simulated frequency for which a circle with the specified value exists.

**Frequency** specifies the spot frequency for circles plotted against a design variable.

**Level Range** defines *Start*, *Stop*, and *Step* for circles plotted for the specified spot frequency.

**Gain** is the value of gain in dB for which gain circles are plotted.

**Noise** is the value of noise figure in dB for which noise circles are plotted.

**Plot buttons and cyclic fields** at the bottom of the form generate the plots. For S, Y, Z, or H parameters, generate plots for ports 1 through 3 by clicking the appropriate button at the bottom of the form. To generate plots for the other ports, use the cyclic fields beside the buttons to specify the output and incident ports, and then click the *S*, *Y*, *Z*, or *H* button to generate the plot.


© 2023 Cadence Design Systems, Inc.

# Circuit Conditions

## Device Operating Conditions

These checkboxes let you highlight components in saturation and in breakdown. When the *Annotate Place* button is pressed, components in breakdown, saturation, or both are highlighted on the schematic with a colored box. The color of the box is chosen by the color cyclic field next to each field.

### *Saturation BJT or Linear MOS*

An instance is highlighted for

- saturation region of BJT if the operating point parameter region=3
- or linear/triode region of MOS/bsim if the operating point parameter region=1

### *Breakdown*

For Spectre breakdown, an instance is highlighted if

- For BJT: If the operating point parameter region=4

**Note:** For the simulator to calculate breakdown or saturation, the appropriate model parameters need to be set.

## User-Defined Conditions

**Enable** uses the cyclic field to select yes or no to enable or disable a condition.

**Color** shows the color with which you want to highlight instances meeting a condition.

**Component** shows the type of component for which you want to create conditions.

**Lower Bound** specifies the lower boundary of a parameter's value.

**Upper Bound** specifies the upper boundary of a parameter's value.

**Parameter** is an operating-point parameter you choose from the cyclic field. The *Lower Bound* and *Upper Bound* values apply to the selected parameter.


© 2023 Cadence Design Systems, Inc.

**and/or** sets Boolean arguments to a condition. When *and* is used, both conditions must be met for an instance to be highlighted. When *or* is used, either condition must be met for an instance to be highlighted. Both operators have the same precedences.

**Add** adds another compound condition to the existing entries in the table. When this button is clicked, a new row is added to the bottom of the table so that a designer can specify another search condition.

**Delete** removes a condition from the table. When this button is clicked the selected entries in the table are removed. You select entries by clicking on a row in the *User Defined Conditions* box.

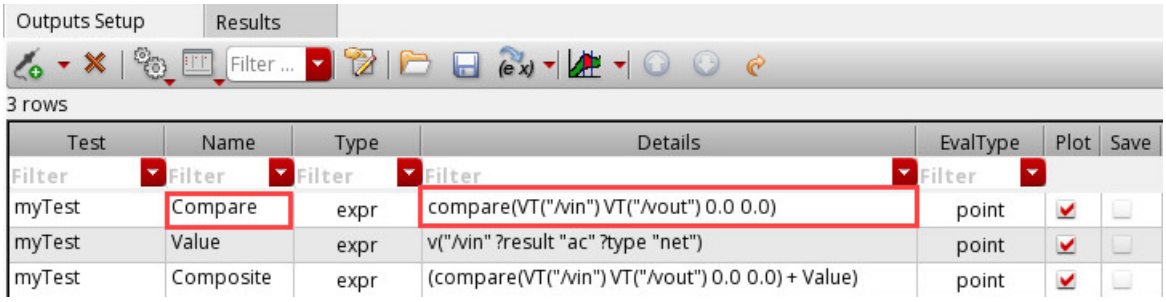
**Change** lets you modify a user-defined condition. You must select the condition before modifying it.

**Clear** lets you clear all the entries from the *User Defined Conditions* box.

### Plotting Associated Signals

You can plot all the signals used in an expression directly from the *Results* tab.

On the *Outputs Setup* tab, consider the `Compare` expression, which compares two waveform signals: `VT("/vin")` and `VT("/vout")`.

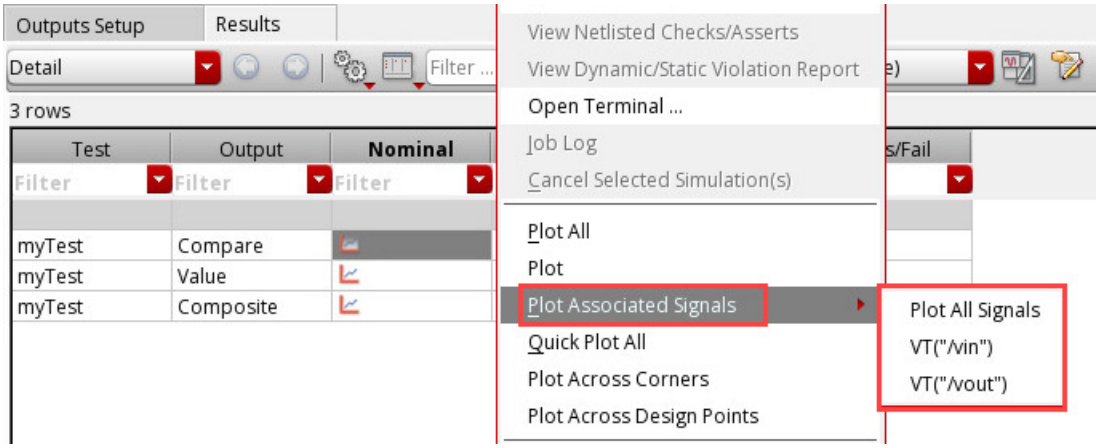



To plot the signals used in the expression, do the following:

1. Run the simulation to evaluate the expression.


© 2023 Cadence Design Systems, Inc.

The results are displayed in the *Results* tab.

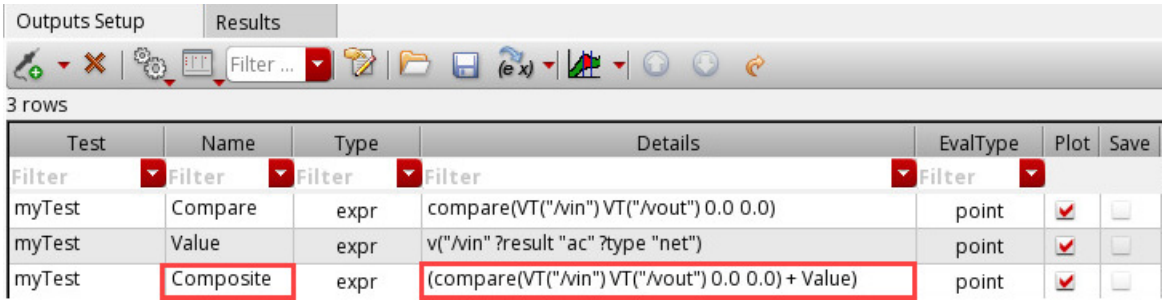



2. On the *Results* tab, right-click the *Nominal* column, and then choose one of the following commands from the context menu:

   - *Plot Associated Signals — Plot All Signals*: Plots all the associated signals used in the expression.
   - *Plot Associated Signals — <Signal 1>*: Plots the first signal used in the expression. For example, `VT("/vin")`.
   - *Plot Associated Signals—<Signal 2>*: To plot the second signal used in the expression. For example, `VT("/vout")`.

   In addition to the associated signals, you can also plot the first-level dependent used in the expression.

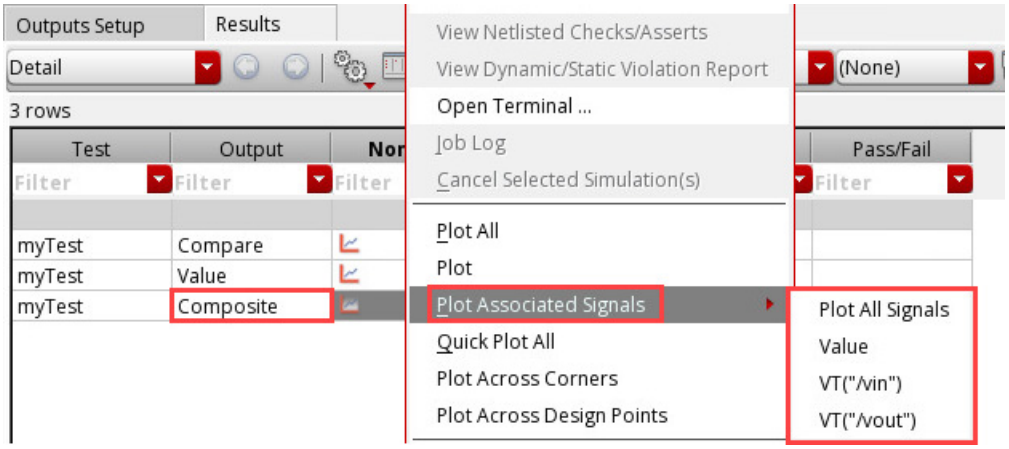
   Consider the `Composite` expression, as shown in the following figure:





© 2023 Cadence Design Systems, Inc.

If you plot the associated signals of the expression, the first-level dependent, `Value`, can also be plotted directly from the Results tab, as shown in the following figure.





© 2023 Cadence Design Systems, Inc.

© 2023 Cadence Design Systems, Inc.

# 10

# Real-Time Tuning in Simulations

The Real-Time Tuning (RTT) assistant in ADE Explorer helps you edit the values of variables and parameters and the simulation results are dynamically updated to reflect the changes that have been made. In this mode, you can run the simulation once and fine-tune your results later by changing the variable values and simultaneously viewing the updated results. The output waveforms are displayed in the Virtuoso Visualization and Analysis XL graph window that can also open up as an assistant in ADE Explorer and displayed to the right of the ADE Explorer or design window. The output waveforms are updated as soon as you change the parameter values in the RTT assistant. You can also change the plotting mode to control the display of waveforms or compare waveforms generated after changing variable and parameter values. The available plotting modes are append, replace, new window, and new subwindow.

**Note:** Real Time tuning does not support string parameters.

This section covers the following topics:


- Opening the Real-Time Tuning Assistant
- RTT Graphical User-Interface
- Working with the RTT Assistant
- Backannotating Variables and Parameters from RTT
- Viewing Results in Virtuoso Visualization and Analysis XL
- Running RTT with Other Simulators
- Viewing Legend in Info-Balloons and ViVA Graph Assistant
- Viewing Waveform Info-Balloons on Schematic



© 2023 Cadence Design Systems, Inc.

## Opening the Real-Time Tuning Assistant

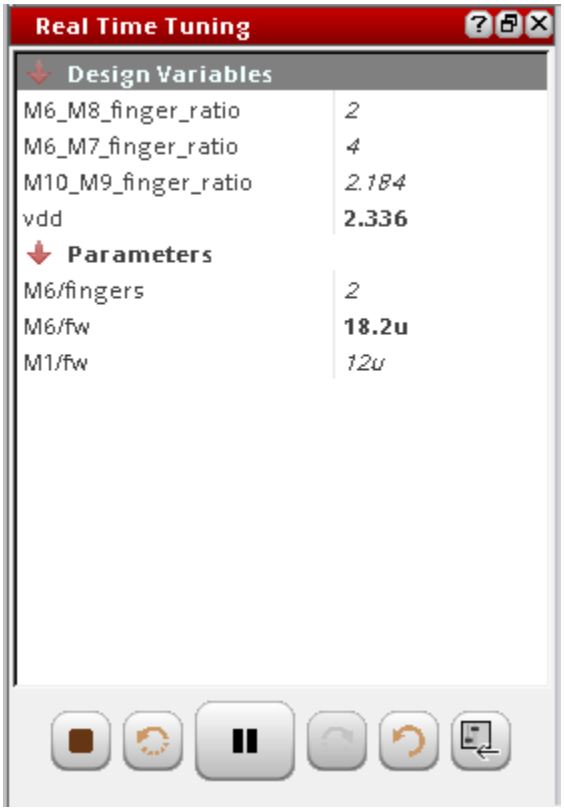
The Real-Time Tuning pane opens as a dockable assistant in schematic and Explorer. By default, when you launch this assistant, it is displayed on the schematic tab. You can click the `maestro` tab to view this assistant in ADE Explorer.

You can open the RTT assistant through one of the following ways:

- From Virtuoso ADE Explorer:
  - Click the button available on the Real Time Tuning toolbar. To hide or show this toolbar, you can right-click anywhere in the toolbar area and choose *Real Time Tuning*.
  - Choose *Launch – Plugins – Real Time Tuning*.
  - Choose *Tools – Real Time Tuning.*
  - Select `Real Time Tuning` in the Workspace toolbar.
- From the Virtuoso ADE Explorer Schematic Window:
  - Click the button available on the Schematic toolbar.
  - Choose *Launch – Plugins – Real Time Tuning*.
  - Select `Real Time Tuning` in the Workspace toolbar.


© 2023 Cadence Design Systems, Inc.

The *Real Time Tuning* assistant appears to the left of the Schematic or Explorer window as shown in the below figure.




*Important*

To open the RTT assistant, you must have at least one output, or, one design variable or parameter defined in the Explorer setup. Otherwise, an error message is displayed and RTT assistant does not open.

**Note:** You cannot tune the design using the Real Time Tuning assistant when the design variable includes cyclic dependency; for example, a=1, b=a+c, and c=a+b.

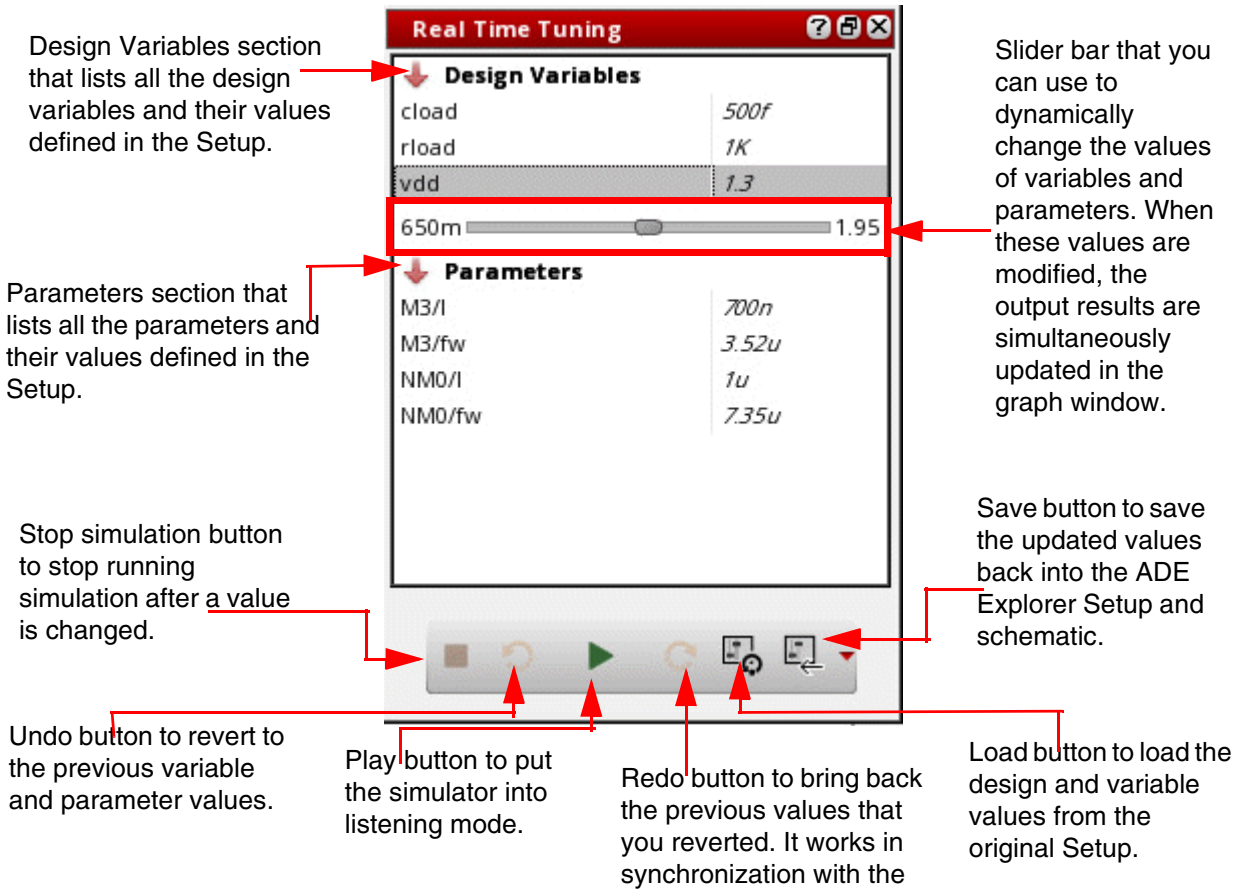
In addition, the specifications are not evaluated when you are in the RTT mode.

For more information about how to work with the RTT mode, see the next sections.


© 2023 Cadence Design Systems, Inc.

## RTT Graphical User-Interface

The below figure demonstrates the RTT graphical user-interface and its main elements:

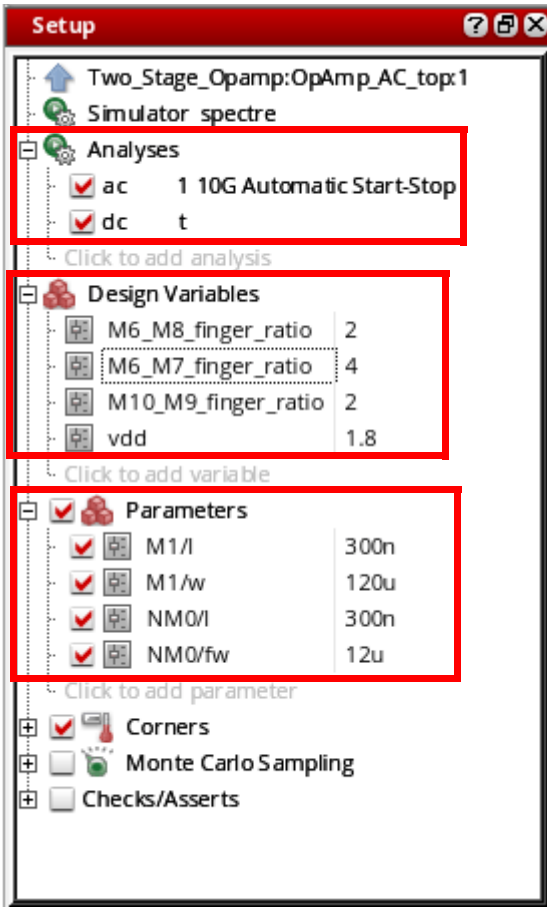




© 2023 Cadence Design Systems, Inc.

## Working with the RTT Assistant

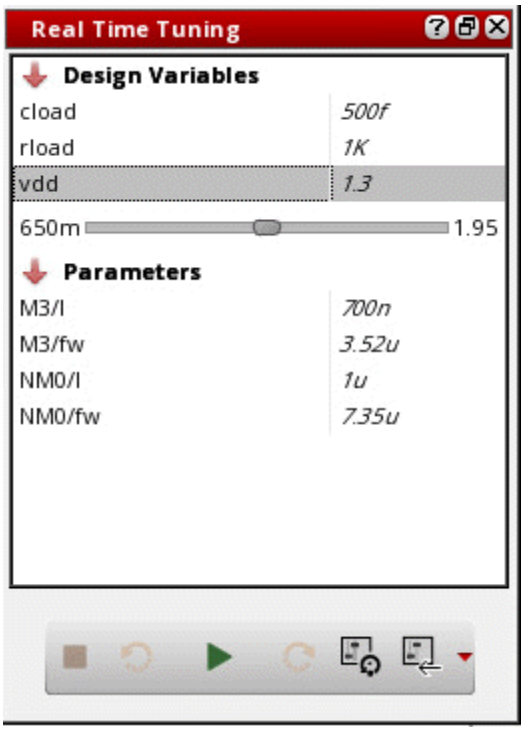
Consider the setup shown in the figure below in which you define the AC and DC analyses, four design variables and four parameters:



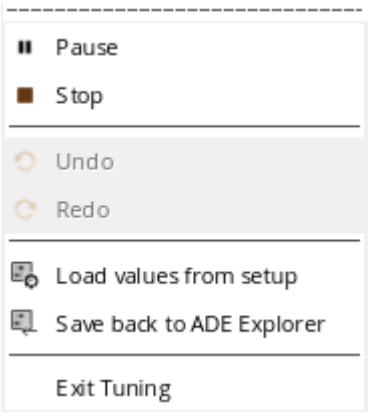


© 2023 Cadence Design Systems, Inc.

Now, open the Real Time Tuning assistant using the steps explained in the Opening the Real-Time Tuning Assistant section. The schematic view appears and Real Time Tuning assistant is displayed with the following values:




The *Tuning* menu also appears when the Real Time Tuning assistant is displayed. This menu includes the following commands:




In the Real Time Tuning assistant, perform the following steps to tune the design and view the updated waveforms instantly in the graph window:


© 2023 Cadence Design Systems, Inc.

1. Click the Play button to put the simulator into listening mode, which means RTT is ready to action according to the design and parameter values you change. When the play button is clicked, an animated spinner appears on this button, which indicates the progress of the simulation run. In addition, the status is also displayed in the status bar at the bottom, indicating the simulation stages, such as simulating, plotting and so on.

   When RTT is opened for the first time, clicking this button first runs the simulation on the values defined in the Setup and plots the results in the graph window.

   **Note:** When the simulator is in listening mode and you perform some changes in the design, if you change the variable and parameter values in RTT assistant, the simulation will now run on the updated design.

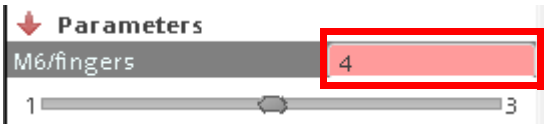
   When the simulation is complete, the button icon is changed to a pause button .

2. In the next step, update the design variables and parameters values through one of the following ways:

   - Use the slider bars available for each variable and parameter, which appear when you hover mouse pointer over a value. These bars display the minimum and maximum range values that can be specified for the selected variable. You can drag the slider in between these values and when you drop, the value of the selected variable is updated. To change the range values, double-click the displayed value and type a new value. By default, the slider bar for only selected variable or parameter appears. To display all the slider bars, set the following .cdsenv variable to `t`:

     ```
     envSetVal("maestro.rtt" "sliderOnMouseOver" 'boolean "nil")
     ```

   - Double-click the value displayed for the selected design variable or parameter and type the new value. You can also type a value greater than the maximum range displayed on the slider bar. The new value is displayed with red background as shown in the figure below. If this specified value is valid, the simulation runs successfully.

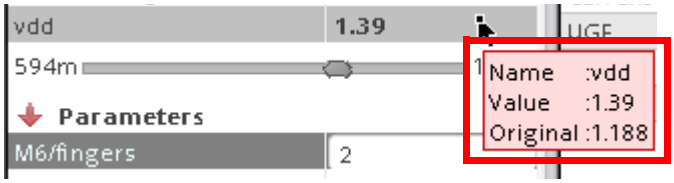
   The values that you change are displayed in bold and the unchanged values are displayed in italics. This makes convenient to distinguish between the new and old values.


© 2023 Cadence Design Systems, Inc.

After changing the value for a parameter or a design variable, if you want to change the value with the one specified in the design, double-click in the value field. A button appears as highlighted in the below figure. Click this button to replace the displayed value with the original value. You can also double-click the specified maximum and minimum values in the slider bar and then specify new values.

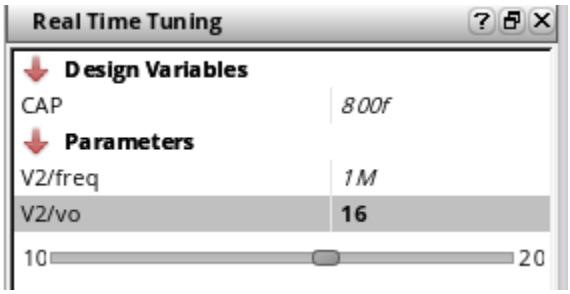
Ensure that the maximum value you specify is greater than the given minimum value. Similarly, minimum value should be less than the given maximum value. Otherwise, an error message appears.

After you change a value, if you place the mouse pointer on the changed value, a tooltip appears that shows the parameter or variable name, its new and the original values.




When you change the variable or parameter value, the updated value is passed to the simulator and an instant simulation run takes place. This simulation is fast because the simulation run does not generate the netlist again.

If the parameter you are tuning is a sweep parameter, then the minimum and maximum values are displayed at the two ends of the slider bar and you can move the slider to select values based on the specified step size. For example, in the below figure, for parameter `V2/vo`, the sweep defined is `10:2:20`. The slider bar shows minimum value as `10`, maximum value as `20`, and you can specify only values that are at the interval of `2`, such as 12, 14, 16, 18. The maximum of 100 points (step values) can be selected through the slider.




**Note:** In the RTT mode, the change in value of temperature also initiates a simulation


© 2023 Cadence Design Systems, Inc.

run.

To set the path of simulation results directory where the intermediate simulation results are saved for the changed values, you can set the following environment variable in CIW or `.cdsinit` file:

```
envSetVal("maestro.rtt" "useCustomizedSimDir" 'string "./")
```

If you do not set this variable, simulation results are saved at the default memory location, `/dev/shm/simulation/<lib>/<cell>/spectre/<view>`.

3. When the simulator is in listening mode, the play button is changed to a pause button. This indicates that you can change only one value at a time and plots are updated in the graph window accordingly. To change multiple values together, click the pause button again. The button icon is changed as shown in the figure below. This indicates that the simulation run is on hold and does not happen every time when you change a value.

Now when you are done with all the changes, click this button again to continue the simulation run.

At any point of time, you can click the stop simulation button ( ) to bring back the play button to its default value and stop running simulation after the values are changed.

To bring back the previous values, use the Undo and Redo buttons available at the bottom of the assistant.

After changing values, if you want to bring back the default values from the previously-saved setup, click the Load values from Setup button ( ).

4. As soon as you update the values in RTT, the updated simulation results are plotted in the Virtuoso Visualization and Analysis XL graph assistant. You can also set the following environment variable in CIW or `.cdsinit` file to open graph as a window in Explorer.

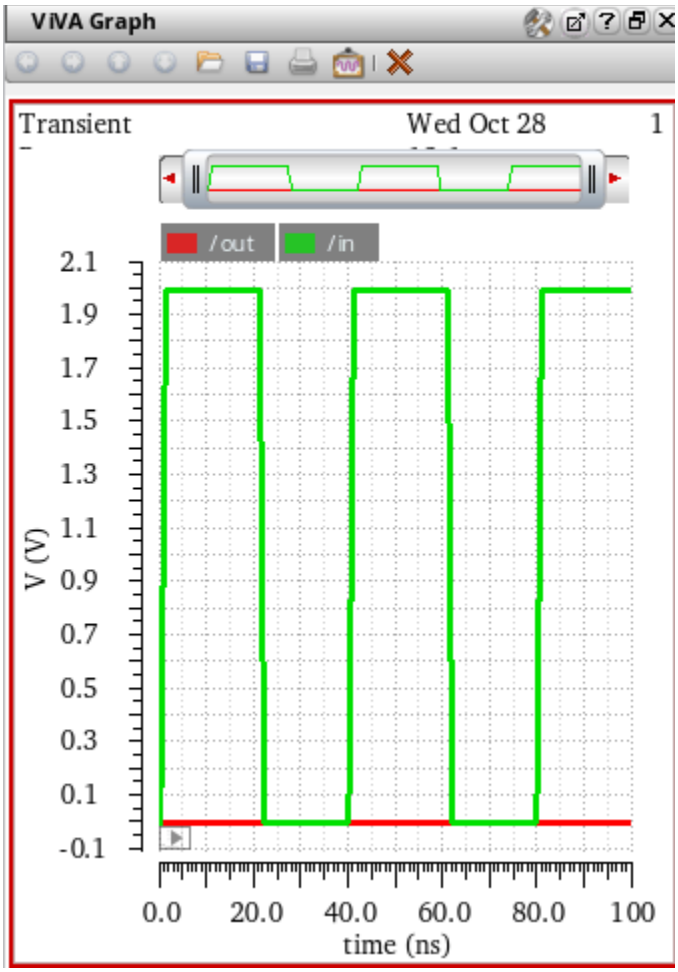
```
envSetVal("viva.application""vivaWindowMode"'string "window")
```

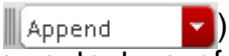
The default value is:

```
envSetVal("viva.application""vivaWindowMode"'string "assistant")
```


© 2023 Cadence Design Systems, Inc.

By default, the graph assistant is displayed at the right of the ADE Explorer window.



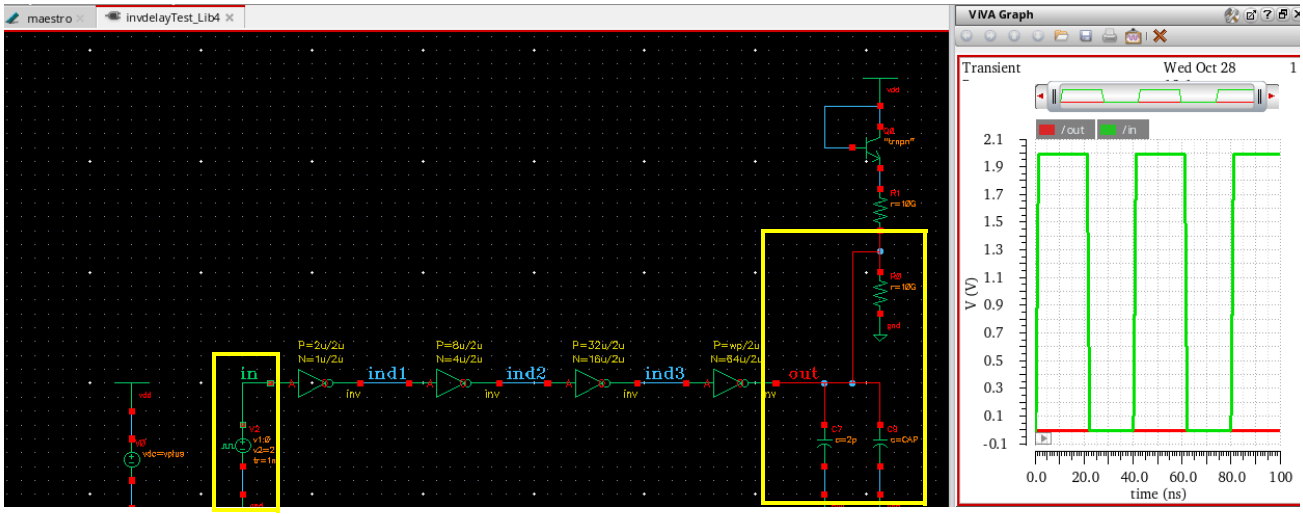

You can use the drop-down menu ( 

) available on the toolbar to change the mode in which you want to display output waveforms. The available modes are Append, Replace, New Window and New Subwindow. For more information about the ViVA graph assistant, see *Virtuoso Visualization and Analysis XL User Guide*.

You can also view the updated simulation results in waveform info-balloons displayed on the schematic design. For more information, see Viewing Waveform Info-Balloons on Schematic on page 879.


© 2023 Cadence Design Systems, Inc.

The nets in the schematic are displayed in the same color as that of the waveform plots in the graph window. See the yellow highlights in the figure below.

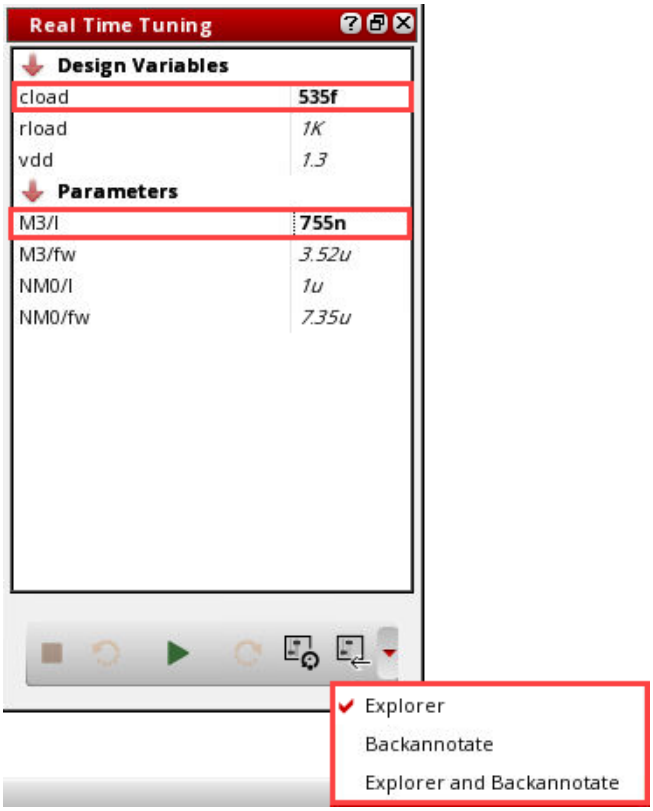




© 2023 Cadence Design Systems, Inc.

## Backannotating Variables and Parameters from RTT

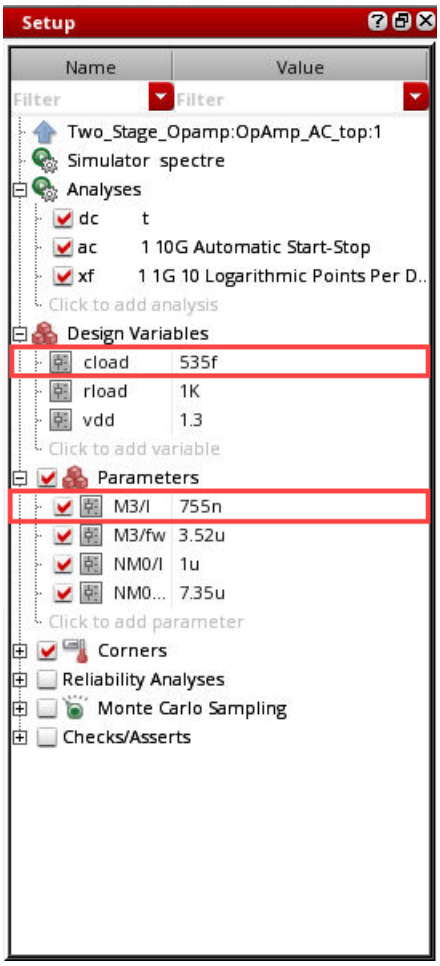
After tuning the design using the *Real Time Tuning* assistant, values of the updated parameter and design variables are displayed in bold, as shown in the following figure..




1. To send these updated values back to ADE Explorer and the schematic, select one of the following options from the *Save back* drop-down menu:


© 2023 Cadence Design Systems, Inc.

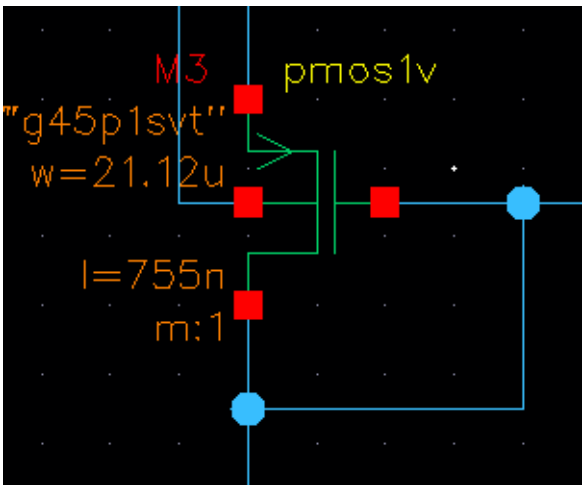
**a.** *Explorer*: updates the values of design variables and parameters in the setup assistant.




**b.** *Backannotate*: Updates the parameter values in the schematic.


© 2023 Cadence Design Systems, Inc.

**Note:** The design variable values are not updated in the schematic.

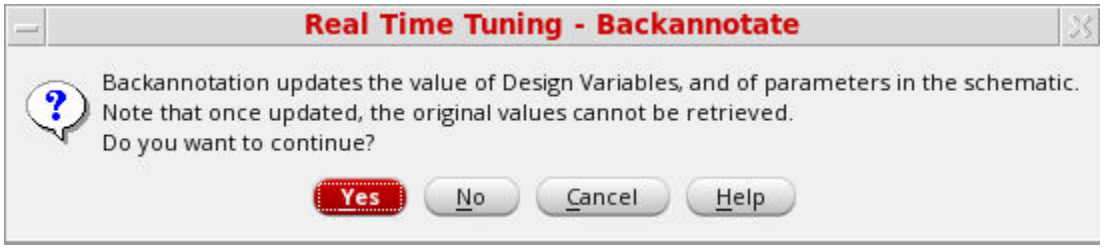



**c.** *Explorer and Backannotate*: Updates the values of the design variables and parameters in the setup assistant, and that of parameters in the schematic.

**Note:** You can control the default selection by setting the value of the environment variable, `saveToMode`.

**2.** Click *Save back.*

The following dialog box is displayed.




**3.** Click *Yes* to confirm the changes.

## Some Useful Points

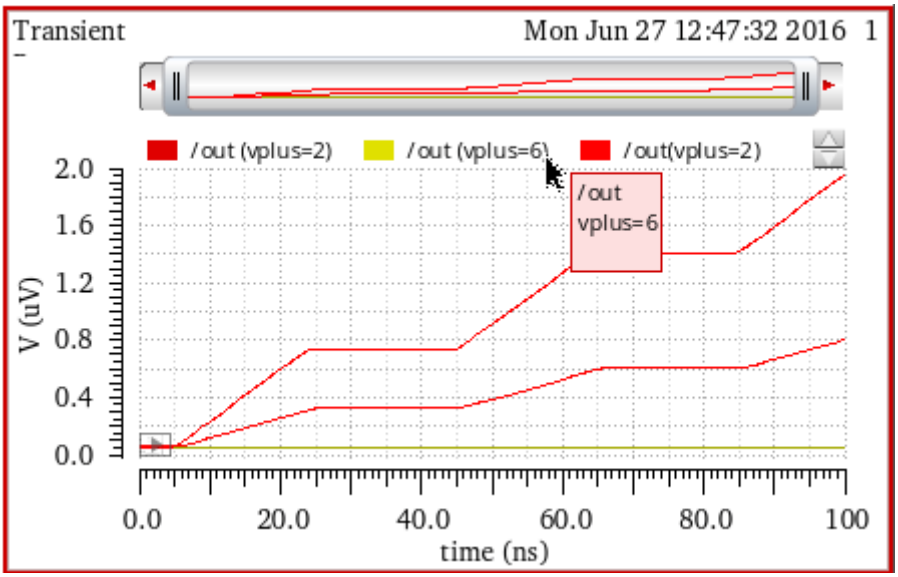
- When you click the Save back button in RTT assistant, only the updated design and parameter values are transferred back into the original setup. RTT does not send the updated results to the ADE Explorer setup. In order to view the updated results in ADE Explorer window, you need to rerun the simulation.


© 2023 Cadence Design Systems, Inc.

- When you are in the RTT mode, the context-sensitive options are not displayed when you right-click a column in the Outputs Setup pane, except for the *Value* column. In addition, the *Results – Save* option is not available in the RTT mode.
- RTT simulation does not evaluate the outputs of type OCEAN.
- If reliability is enabled when you use RTT assistant to modify the design, the Spectre batch mode is used to run the RelXpert simulation and the following message is displayed:

  ```
  *Info* Spectre SKI interactive mode is disabled in RTT because RelXpert setting
  is enabled. Automatically setting the run mode to batch.
  ```

- In the RTT mode, the trace legend displays the parameter values along with the trace name. The trace legend for the waveform info-balloon on schematic also works in the similar way. When the legend is positioned at left or top, tooltip on the trace name in legend displays the trace name, parameter names and values in separate lines, as shown in the figue below.

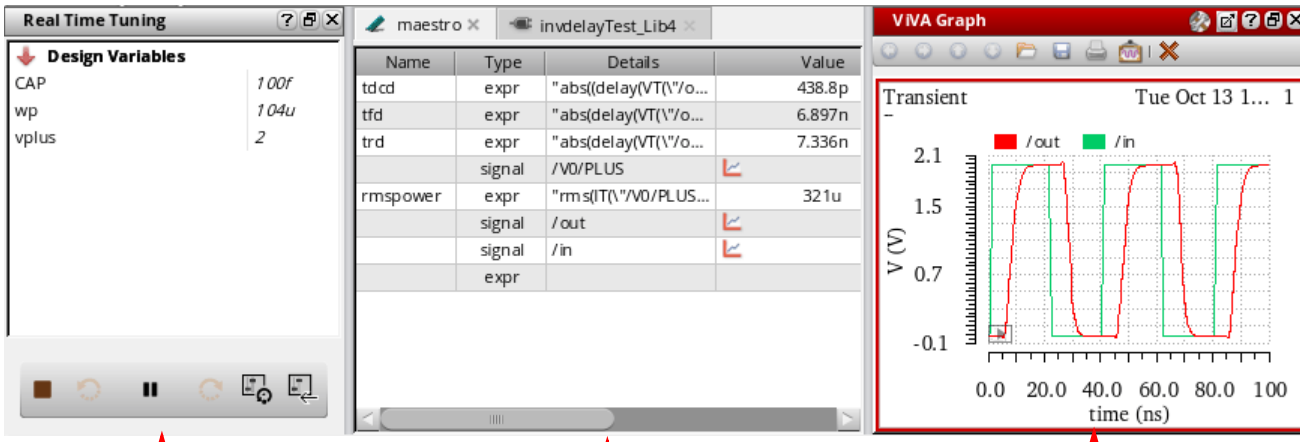




© 2023 Cadence Design Systems, Inc.

## Viewing Results in Virtuoso Visualization and Analysis XL

When the simulation run is complete, the plots are updated in Virtuoso Visualization and Analysis XL Graph. When the `vivaWindowMode` environment variable is set to `Assistant`, the ViVA graph assistant appears to the right of the Explorer or design window.



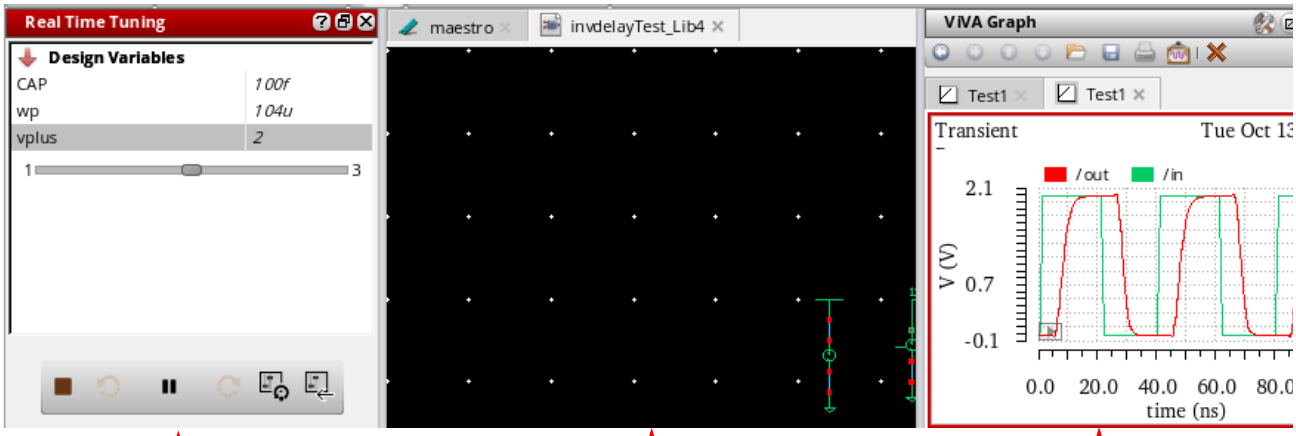

Real Time Tuning assistant

Outputs Setup or Results Pane

Virtuoso Visualization and Analysis XL Graph opened as a dockable assistant and displays updated waveform plots when a value is changes in RTT


© 2023 Cadence Design Systems, Inc.

When you are working with the RTT mode in Schematic Editor, the Virtuoso Visualization and Analysis XL graph appears as shown in the figure below:




Real Time Tuning assistant

Schematic design

Virtuoso Visualization and Analysis XL Graph opened as a dockable assistant and displays updated waveform plots when a value is changes in RTT

After changing the values in RTT assistant, you can zoom in a graph portion to view and analyze results at specific points on the graph. Now, if you change a parameter in the RTT assistant, the graph is updated as per the new values and is displayed with the same zoom settings that you set previously.

For information about the graphical-user-interface of ViVA Graph assistant, see ViVA Graph Assistant on page 83.

### Running RTT with Other Simulators

When you use Spectre from MMSIM14.1ISR3 release or onwards (with or without APS mode enabled), the RTT assistant works in the interactive mode. The netlist and full simulation run take place only for the first time when you click the *Play* button. After that when you change the parameter or variable values in the assistant, the instant simulation run takes place and the corresponding output waveforms are simultaneously updated in the ViVA XL Graph assistant or in the waveform info-balloons on schematic.

When you use a Spectre version older than MMSIM14.1ISR3 release or use any other simulator, such as AMS, HSPICE, UltraSim, the RTT runs in a batch mode. This means, every time when a value is changed, a new netlist is generated and full simulation run takes place.


© 2023 Cadence Design Systems, Inc.

Due to this reason, the simulation is slower and you may notice a delay in the display of output waveforms in the ViVA XL Graph assistant and info-balloons on schematic. A warning message is also displayed informing that maximum tool performance can be achieved only if you use Spectre from MMSIM14.1ISR3 release or onwards.

**Note:** If you are using Spectre version prior to MMSIM14.1ISR3, the RTT simulations are run in the batch mode.


© 2023 Cadence Design Systems, Inc.

# Viewing Waveform Info-Balloons on Schematic

After a simulation run is complete, the output expressions are evaluated and resultant waveforms are plotted in the Virtuoso Visualization and Analysis XL graph. You can also view these waveforms in the schematic design in the form of waveform info-balloons or thumbnails.

This section covers the following topics:

- Enabling Waveform Info-Balloons
- Moving and Resizing Waveform Info-Balloons
- Pinning Waveform Info-Balloons
- Changing Analysis Type for Waveform Info-Balloons
- Using Waveform Info-Balloon Context-Sensitive Menu Options
- Viewing RTT Simulation Results in Waveform Info-balloons

Video

The Waveform Thumbnails video is available on Cadence Online Support that demonstrates how to use and configure waveform thumbnails to let you view simulation plots directly on the schematic canvas.

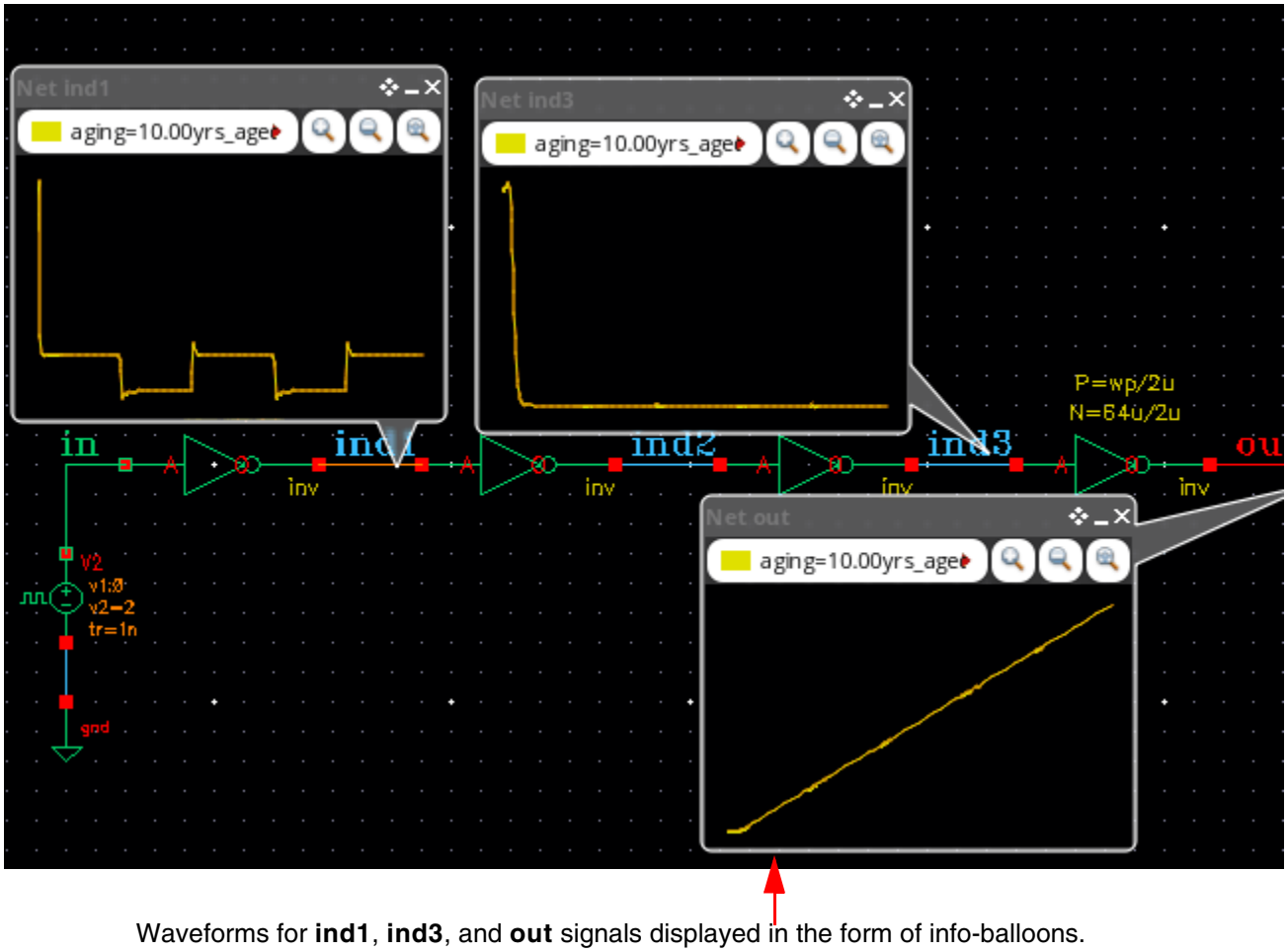
### Enabling Waveform Info-Balloons

To enable the display of waveform info-balloons in schematic, do one of the following:

- Choose *View – Info Balloons*.
- Press the bindkey `Ctrl+B`.
- In the schematic window, click the *Show/Hide Balloons* drop-down ( ) and select *Waveforms*.


© 2023 Cadence Design Systems, Inc.

After the simulation run is complete, the info-balloons appear when you place the pointer over a terminal or an instance associated with the waveform, as shown in the figure below.




Waveforms for **ind1**, **ind3**, and **out** signals displayed in the form of info-balloons. Notice that the legend is displayed in the balloons by default and includes a drop-down list.

Notice that info-balloons in the above figure display a drop-down for the legend inside the trace. This drop-down includes a list of waveforms generated for the sweep values for each leaf. When the simulation includes sweep variables, the legend drop-down displays a list of waveforms generated for all sweep values. This drop-down also maintains a history of waveforms generated for the previous simulation runs.

In addition, the Zoom in by 2, Zoom out by 2, and Zoom Fit buttons are displayed in the info-balloon.

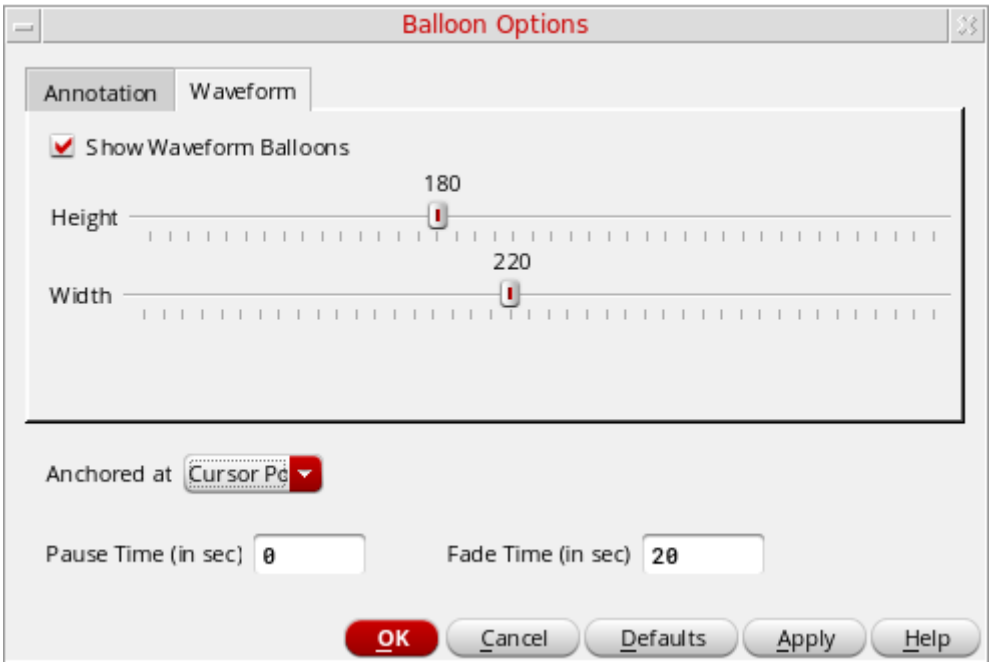

© 2023 Cadence Design Systems, Inc.

### Moving and Resizing Waveform Info-Balloons

You can move or resize only pinned waveform info-balloons. To move and resize the waveform info-balloon, do one of the following:

- Drag the waveform info-balloons using the pointer to change their current position and increase or decrease their size.
- Click the *Balloon Setup Form* button ( ) on the toolbar.
- Click the *Annotation Setup* form button ( ) and choose *Setup – Balloon Setup*.

  The Balloon Options form appears as shown in the figure below:




  In this form, click the *Waveform* tab and then select the following options to display, hide, and resize the waveform info-balloons:

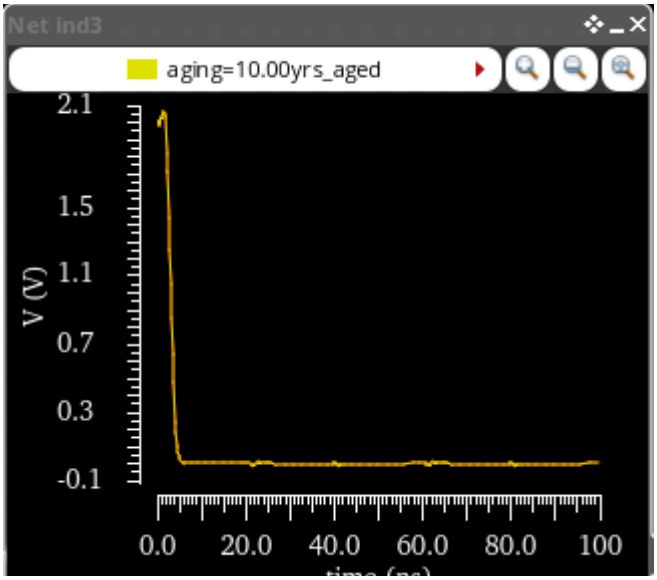
  - *Show Waveform Balloons*—Enable the display of waveform info-balloons when you place the poinyter over an instance.
  - *Height*—Drag the slider bar to set the maximum height of the balloon that can be displayed on the schematic canvas. The default height is `160`.
  - *Width*—Drag the slider bar to set the maximum width of the balloon that can be displayed on the schematic canvas. The default width is `200`.
  - *Anchored At*—Select the position at which the info-balloon is to be displayed when you place the cursor on any instance on the schematic. The available options are:


© 2023 Cadence Design Systems, Inc.

`Cursor Position`, `Top Left Corner`, `Top Right Corner`, `Bottom Left Corner`, and `Bottom Right Corner`.

- *Pause Time (in sec)*—Specify the time taken for an info-balloon to appear when you place the pointer over an instance in the schematic.
- *Fade Time (in sec)*—Specify the time taken for an info-balloon to disappear when you place the pointer over an instance in the schematic.

The axis scale and labels appear when a waveform info-balloon is resized, as shown in the figure below. If the info-balloon size is set to `198x205` or larger, the axes are displayed by default.




### Pinning Waveform Info-Balloons

By default, you can view only one info-balloon at a time because they are displayed only when you place the pointer over an instance. If you want to view and analyze more than one waveform in the design, you can pin the selected info-balloons so that they continue to display.

To pin the info-balloon for a particular instance, do one of the following:

- Click the Pin button ( ) displayed on the header of waveform info-balloon.
- Press bindkey `Shift+B`.
- When you move the mouse pointer over a net, if the pointer changes, you can left-click to pin the info-balloon at the selected position.


© 2023 Cadence Design Systems, Inc.

### Clearing Pinned Waveform Info-Balloons

To clear waveforms displayed in the pinned info-balloons:

➡ Click the *Clear Waveform Balloons* button ( ) on the Annotations toolbar in schematic window.

After you click this button, all the waveforms displayed in the pinned waveform info-balloons are removed and the info-balloons appear blank.

### Changing Analysis Type for Waveform Info-Balloons

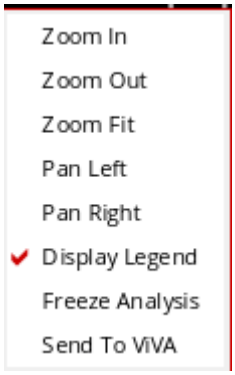
The waveform info-balloons display the plots for only one analysis at a time. To change the type of analysis, select the required analysis type in the drop-down list available on the Annotations toolbar.

The waveform info-balloons for only the selected analysis are displayed in the schematic.

For example, if you choose `ac` analysis type in the drop-down list, the new info-balloon you add shows the plots for AC analysis and all pinned balloons are also changed to show the plots for AC analysis. You can use the *Freeze Analysis* option to lock the analysis type, which is explained in the next section.

### Using Waveform Info-Balloon Context-Sensitive Menu Options

When you right-click anywhere in the waveform info-balloon, the following context-menu options appear:




■ *Zoom In*—Zooms in to the trace.


© 2023 Cadence Design Systems, Inc.

- *Zoom Out*—Zooms out from the zoomed-in trace.
- *Zoom Fit*—Fits the trace to the graph area.
- *Pan Left*—Pans the trace to the left-side of the graph.
- *Pan Right*—Pans the trace to the right-side of the graph.
- *Display Legend*—Controls the display of legend in the info-balloon. It is enabled by default.
- *Freeze Analysis*—Locks the analysis type for the traces displayed in waveform info-balloon. When you click the *Freeze Analysis* option, a lock symbol ( ) appears at the topmost left corner of the info-balloon, which indicates that the current trace is locked for the selected analysis type and does not update if you choose a different analysis. The traces that do not have the *Freeze Analysis* option enabled are updated with a change in the analysis type.

  **Note:** This option works only for the pinned waveform info-balloons.
- *Send To ViVA*—Sends the trace displayed in the waveform info-balloon to Virtuoso Visualization and Analysis XL window.

### Viewing Legend in Info-Balloons and ViVA Graph Assistant

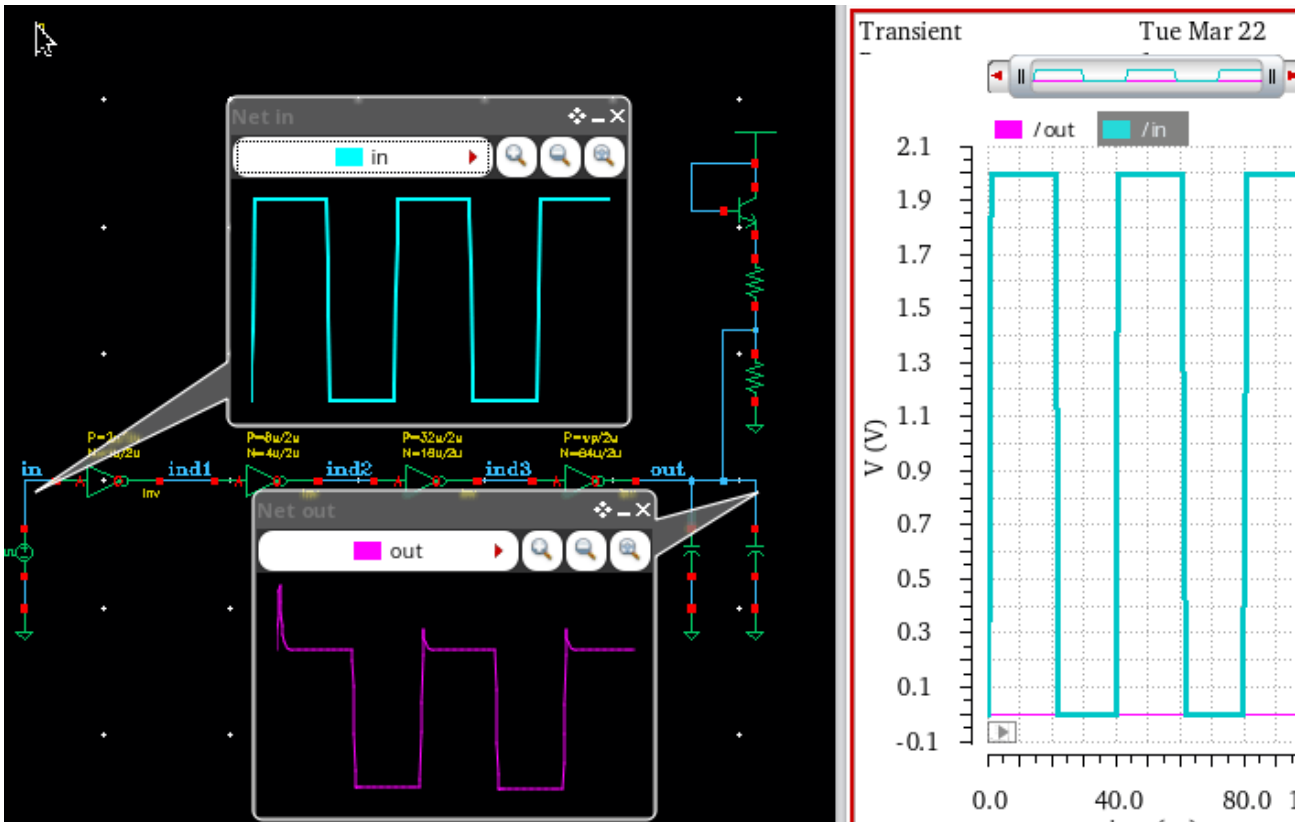
When you run a single-point simulation, the legend on the waveform info-balloons display the name of the net. If you run a multi-point simulation, the legend also display the sweep values along with the net names.

Consider an ADE Explorer setup in which you have the following three design variables:

- `CAP=230u`
- `wp=100u`
- `vplus=2.8u`


© 2023 Cadence Design Systems, Inc.

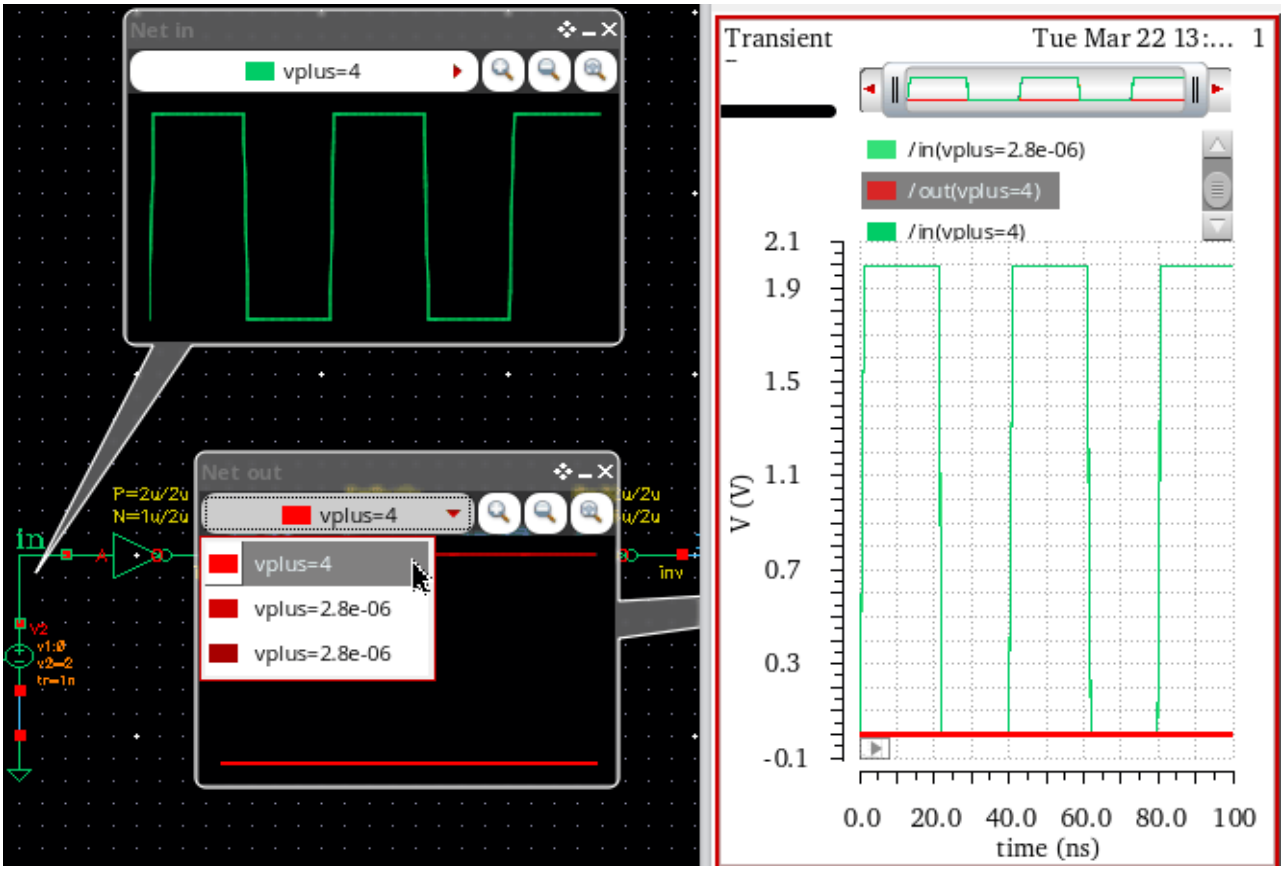
Now, if you run a simulation, the output waveforms are plotted in the ViVA Graph assistant and also displayed in the waveform info-balloons, as shown in the figures below. The color of waveforms in the info-balloon is same as that of the color in the ViVA Graph assistant.




Now, if you change a variable value, say `vplus` to `3.0`, and rerun the simulation, the waveforms are updated in the ViVA Graph assistant and the pinned waveform info-balloons. When the plotting mode is set to append, notice that the legend drop-down in the ViVA Graph assistant and info-balloons display both the new (`3`) and previous (`2.8`) values of the changed variable, `vplus`. The other two design variables remain unchanged, and therefore, are not displayed in the legend.


© 2023 Cadence Design Systems, Inc.

If you specify a sweep for the `vplus` design variable (`vplus=0:2:6`), the waveform info-balloons display the waveforms generated for sweep values and the name of the net, as shown in the figure below.



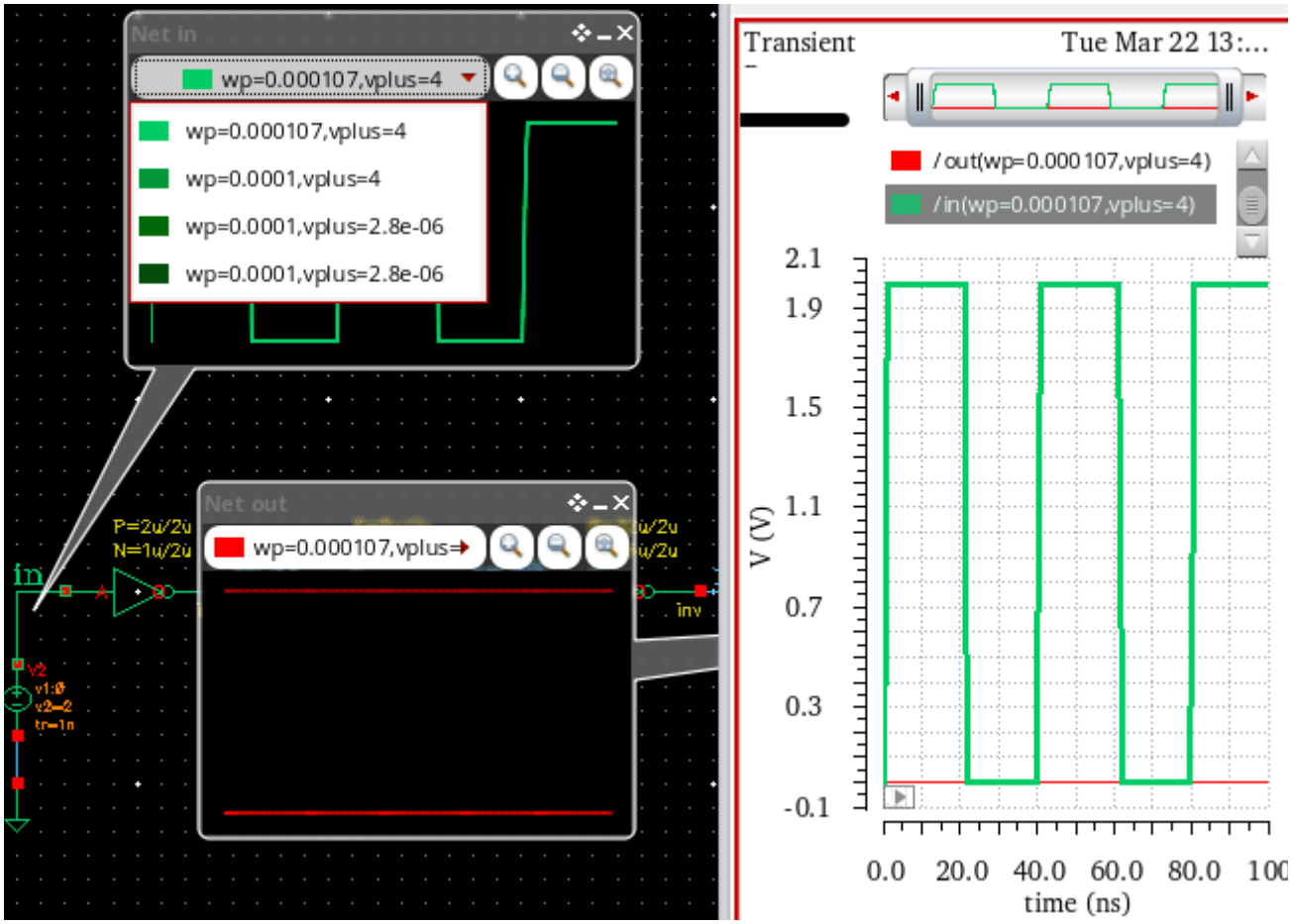

The color of waveforms in the info-balloon is same as that displayed in the ViVA Graph assistant. Also, if you select a waveform in the info-balloon, the same waveform is highlighted in the ViVA Graph assistant. For example, in the above figure, the `out` signal at `vplus=4` is highlighted in the waveform info-balloon and ViVA Graph assistant.

### Viewing RTT Simulation Results in Waveform Info-balloons

After updating the design variables and parameter values in the RTT assistant, you can view the output waveforms instantly in the ViVA Graph assistant. If the waveform info-balloons are enabled in the schematic design and you change a variable and parameter value using RTT assistant, the plots in waveform info-balloons are also updated to reflect the change in output waveforms. These updated waveforms in info-balloon are plotted in different colors in append mode. The highlighted waveform color shows that this waveform is generated by the recent simulation run.


© 2023 Cadence Design Systems, Inc.

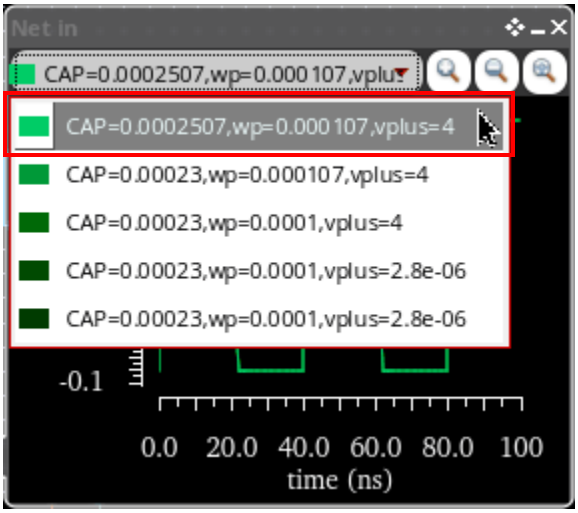
When you tune the design in RTT, the legend drop-down in the info-balloon includes the previous and new values of the variables you changed. For example, consider an example in which you have three design variables, `CAP=230u`, `wp=100u`, and `vplus=2.8u`, in the ADE Explorer setup. When you click the *Play* button for the first time, a simulation run takes place. The legend of the output waveforms in the ViVA graph assistant and info-balloons display only the net name. Now, if you change a variable, say `wp` to `107u` and `vplus` to `4`, the info-balloon legend drop-down includes both the old and new values as shown in the below figure:




The third variable, `CAP`, is not displayed in the legend because its values are not bechanged. Now, if you change the value of other variable, say `CAP`, the legend displays the previous and new values of all the changed variables. In addition, when you can select one of the


© 2023 Cadence Design Systems, Inc.

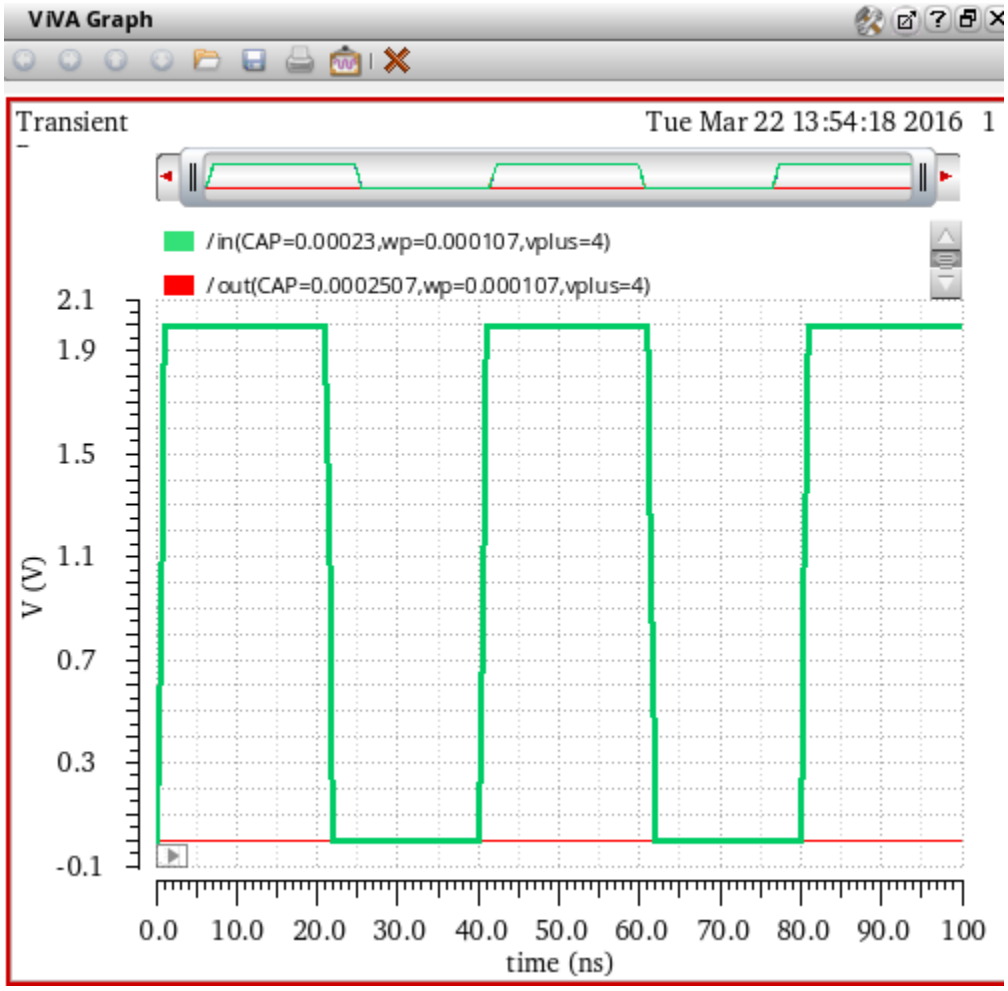
waveforms in the legend drop-down list, the corresponding plots are selected in the info-balloon and in ViVA Graph assistant.




Notice that the color of waveforms from the previous runs becomes dimmer every time when a new simulation is run. The waveform with the brightest color indicates the latest plot. Alternatively, you can set the `dimSameColor` environment variable to `nil` to display these waveforms in different colors. The default value of this variable is `t`.


© 2023 Cadence Design Systems, Inc.

Similarly, the ViVA Graph assistant legend also displays the values for only changed variables (see the figure below).




**Points to Note:**

- The plots for the updated results are displayed only when the waveform info-balloons are pinned.
- The waveform info-balloons are not currently supported for bus terminals. If you try to add a waveform info-balloon on a bus terminal, an information message is displayed in CIW.


© 2023 Cadence Design Systems, Inc.

© 2023 Cadence Design Systems, Inc.

# 11

# Viewing Results

This chapter covers the following topics:


- Viewing the Simulation Results on page 892
- About Results Views on page 900
- Backannotating from ADE Explorer Results on page 918
- Re-evaluating Expressions and Specifications on page 920
- Viewing Specification Results in the Results Tab on page 921
- Comparing S-Parameter Data on page 923
- Working with Datasheets on page 931



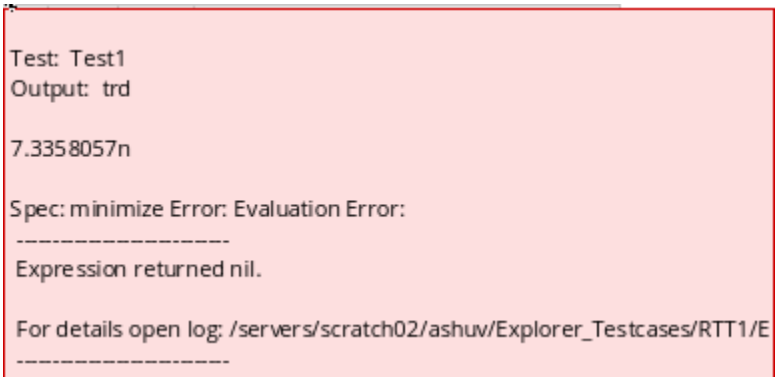
© 2023 Cadence Design Systems, Inc.

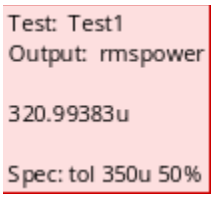
# Viewing the Simulation Results

After a simulation run is complete, the results are displayed in the *Value* column of the Output Setup.

| Name | Type | Details | Value | Plot | Save | Spec |
|---|---|---|---|---|---|---|
| | signal (I) | /V1/PLUS | | ☑ | ☑ | |
| Current | expr | abs(IDC("/V1/PLUS")) | 1.163m | ☑ | ☐ | < 1.5m |
| UGF | expr | cross(dB20(mag(VF("/OUT"))) 0 ... | 481.9M | ☑ | ☐ | > 533M |
| Gain | expr | value(dB20(mag(VF("/OUT"))) 1) | 47.39 | ☑ | ☐ | > 44 |
| Voffset | expr | (VDC("/inm") - VDC("/inp")) | 3.011m | ☑ | ☐ | range -10m 10m |
| CMRR | expr | (value(dB20(mag(VF("/OUT"))) 1... | 48.31 | ☑ | ☐ | |
| GainVsupply1 | expr | value(db(getData("/V1" ?result "... | -6.87 | ☑ | ☐ | |
| PSRR | expr | (value(dB20(mag(VF("/OUT"))) 1... | 54.26 | ☑ | ☐ | |
| GainCm | expr | value(db(getData("/Vcm" ?resul... | -916.2m | ☑ | ☐ | |

The *Value* column includes the signal icon for the output plots and scalar values for results. If the results pass, the background color in the *Value* column is green. If the results fail, it is displayed in red color. You can mouse-hover in this column to view additional information about the results. For example, the following tooltips appear for the failed and passed results:



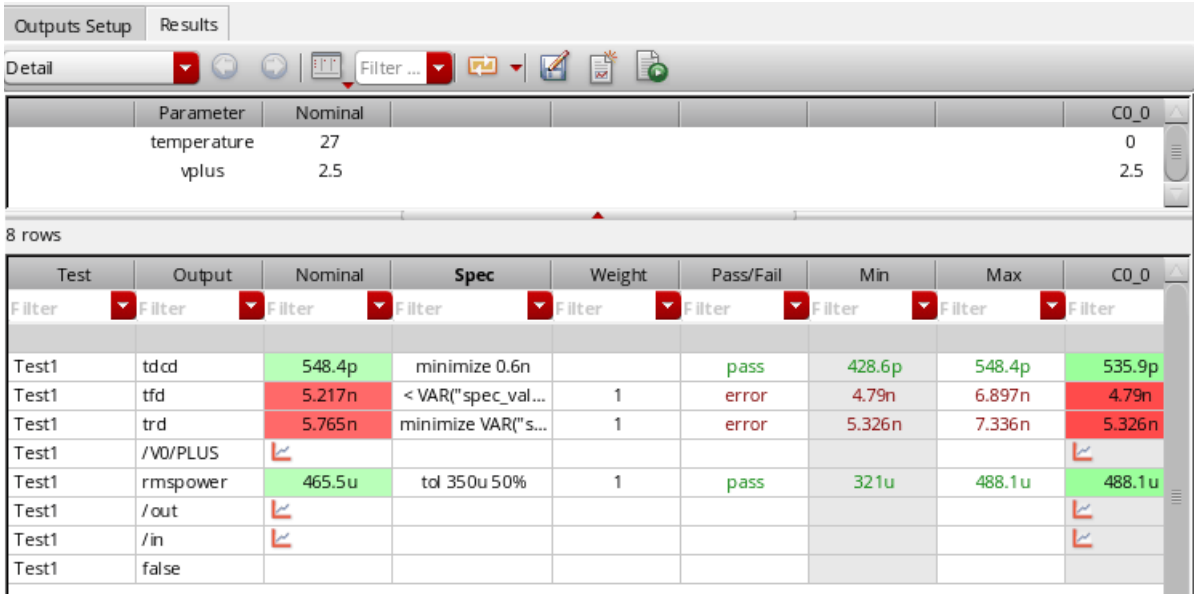

**Note:** You can also sort the results based on their types by clicking the arrow button in the header of the Value column.

The Results tab is displayed when you run a simulation with the following features:

- Multi-point simulation with sweep values defined for parameters or corners


© 2023 Cadence Design Systems, Inc.

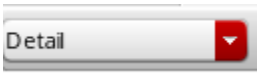
- Monte Carlo Sampling
- Reliability Analysis
- Checks/Asserts analyses




The *Nominal* column in the Results tab displays the simulation status such as pending, running, error codes, and so on, and the text foreground color is changed based on error code of the result. A waveform icon is displayed in this column when the output result is a wave.

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 by using the *Select the results view* drop-down list.




You can export results to HTML or a comma-separated values file.

**Note:** If you run a simulation in ADE Assembler, the simulation results from ADE Assembler cannot be saved or selected in ADE Explorer. However, if you have already viewed results in ADE Assembler and you move to ADE Explorer with the results still open, you can perform post-processing operations in ADE Explorer, such as plotting, printing, and annotation.


© 2023 Cadence Design Systems, Inc.

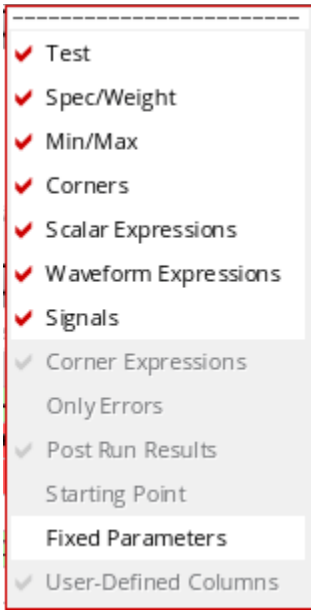
## Hiding and Showing Data on the Results Tab

The following data appears on the Results tab for the Detail View.

| Column | Data/Content |
|---|---|
| Point | Number of the design point for which results are displayed. |
| *Test* | Test name, top of expandable tree of outputs |
| *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 |
| *Spec* | Specification information |
| *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 |
| *Min* | Minimum value for the specification |
| *Max* | Maximum value for the specification |
| `CornersName` | Measured value or waveform at each corner |


© 2023 Cadence Design Systems, Inc.

You can show or hide columns and rows displayed on the Results tab by using the (*Configure what is shown in the table*) drop-down on this tab. When you click this button, ADE Explorer shows the following menu.




You can select the column name from this drop-down that you want to hide.

If you run a Monte Carlo simulation, the following options are listed in the *Configure what is shown in the table* drop-down list:

© 2023 Cadence Design Systems, Inc.

## Hiding Test Details

Test details appear in the *Test* column on the Results tab. To hide test details, click the *Test* option. The *Test* column disappears from the Results tab.

## Hiding Specification Details

Specification details appear in the *Spec* and *Weight* columns on the Results tab. To hide specification details, click the *Spec/Weight* option.

## Hiding Corner Results

Corner results appear in columns on the Results tab, each result in its own column. To hide corner results columns, click the Corners option.


© 2023 Cadence Design Systems, Inc.

## Hiding Signals

Signals appear as rows with the icon in the *Nominal* column and the columns for corners on the Results tab. To hide signals, click the *Signals* option.

The signal rows disappear from the Results tab. If you have any measured values for expressions, those are the only values that appear.

**Note:** This command is available only in the `Detail`, `Detail - Transpose`, and `Optimization` result views.

## Hiding Measured Result Values

Measured result values appear in the *Nominal* column and the columns for corners on the Results tab.

To show or hide these measured result values, click one of the following options:

- *Scalar Expressions*: Shows or hides results of the expressions that evaluate to scalar results.
- *Waveform Expressions*: Shows or hides results of the expressions that evaluate to waveform results.

**Note:** These commands are available only in the `Detail`, `Detail - Transpose`, and `Optimization` result views.

## Hiding Minimum and Maximum Values

The minimum and maximum values appear in the *Min* and *Max* columns on the Results tab. To hide minimum and maximum values, click the *Min/Max*. option. The *Min* and *Max* columns disappear from the Results tab.

## Showing Only Errors

To show errors only, click the *Only Errors* option.

## Filtering Results Using Column Filters

When the number of outputs are large, you can apply filters to selectively display the results you are interested in viewing. For more details, refer to the section *Filtering Results on the Results Tab* in the Virtuoso ADE Assembler User Guide.


© 2023 Cadence Design Systems, Inc.

Video

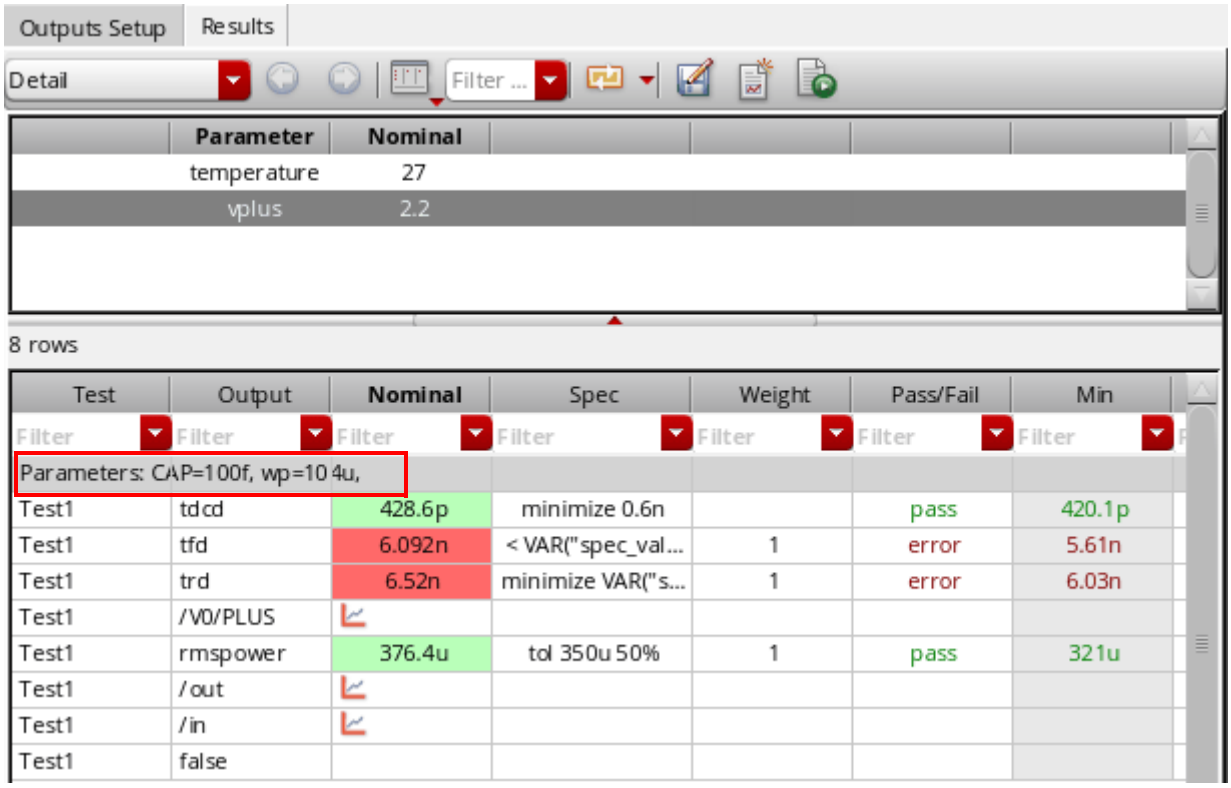
You can view a video demonstration for this feature at Filtering Data on the Outputs Setup and Results Panes.

## Showing the Starting Point

The reference point specified for Monte Carlo Sampling run is displayed as the starting point on the Results tab. To show the starting point on the top of the Results tab, click the *Starting Point* option.

## Showing Fixed Parameters 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, choose *Fixed Parameters*.

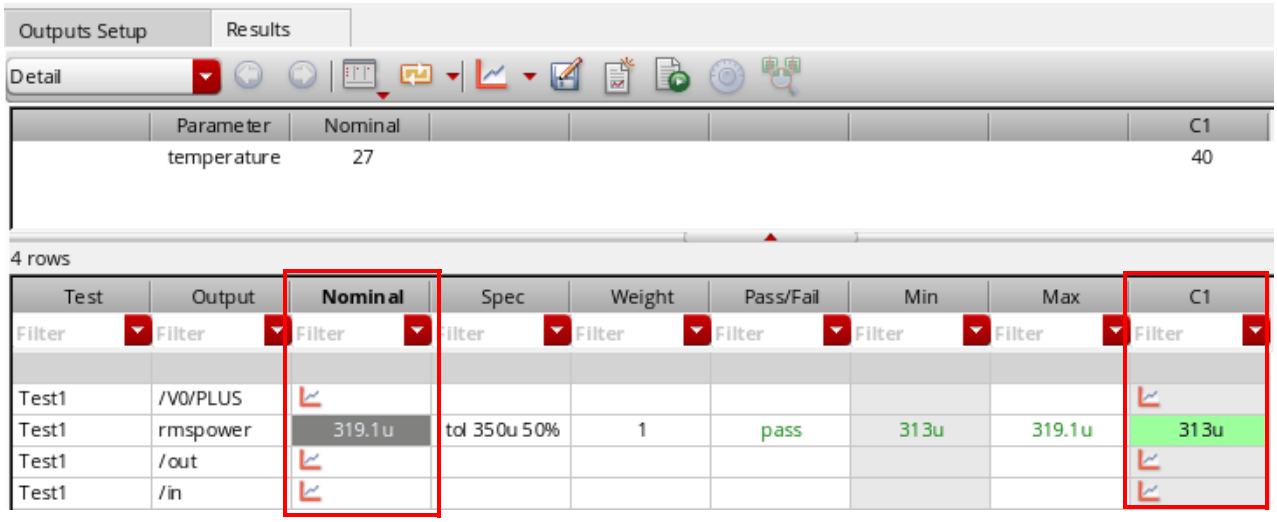




© 2023 Cadence Design Systems, Inc.

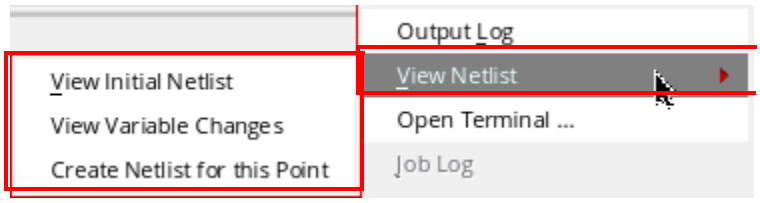
## Viewing Netlist for Results

To view the netlist generated during a simulation run for a nominal point, right-click in the column for nominal point and choose *View Netlist*. The nelist (`input.scs` file for Spectre) is displayed.




However, for a multi-point interactive simulation run, the *View Netlist* option for all points includes the following sub-options that help you view the changes made during the simulation run:

- *View Initial Netlist*—Opens the netlist file. It is the same as the *View Netlist* option for the nominal point.
- *View Variable Changes*—Opens the `spectre_change.log` file that displays the variables changed for the current point with respect to the previous point.
- *Create Netlist for this Point*—Creates netlist for the selected point only.




**Note:** The interactive mode does not support statistical corners, check/asserts, relxpert, and spice outputs. Therefore, simulations involving them are automatically run in batch mode. In addition, in interactive mode, the netlist is created only once and includes all corners or sweeps within the altergroups. Any further changes to sweeps or corners are directly passed onto the simulator.


© 2023 Cadence Design Systems, Inc.

# About Results Views

By default, the Results tab shows the Detail view of results. You can use the *Select the results view* drop-down list to choose any other view. The views available in ADE Explorer are:

- Detail
- Detail - Transpose
- Optimization
- Status
- Yield
- Checks/Asserts
- Capacitance

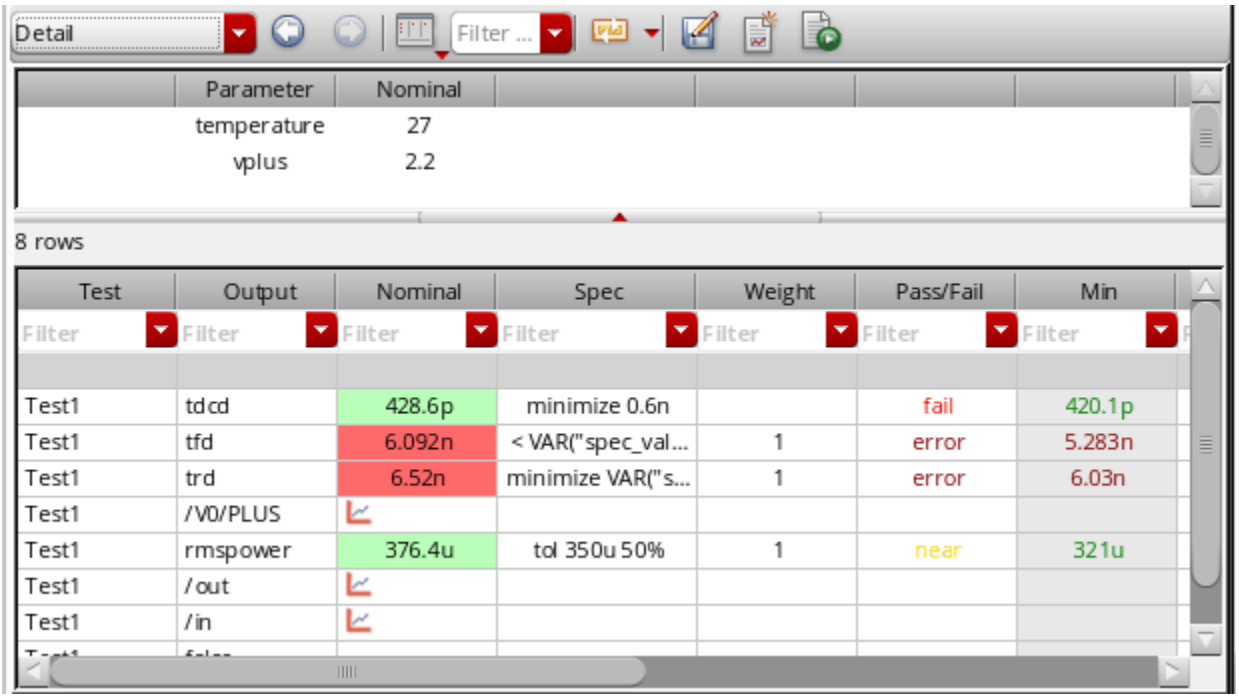
Every run mode has a default result view mapped to it. For example, the default result view for the Monte Carlo Sampling run mode is the Yield view.

## Detail

Displays the detailed results.


© 2023 Cadence Design Systems, Inc.

For more information about the columns in the *Detail* view, see the previous section.

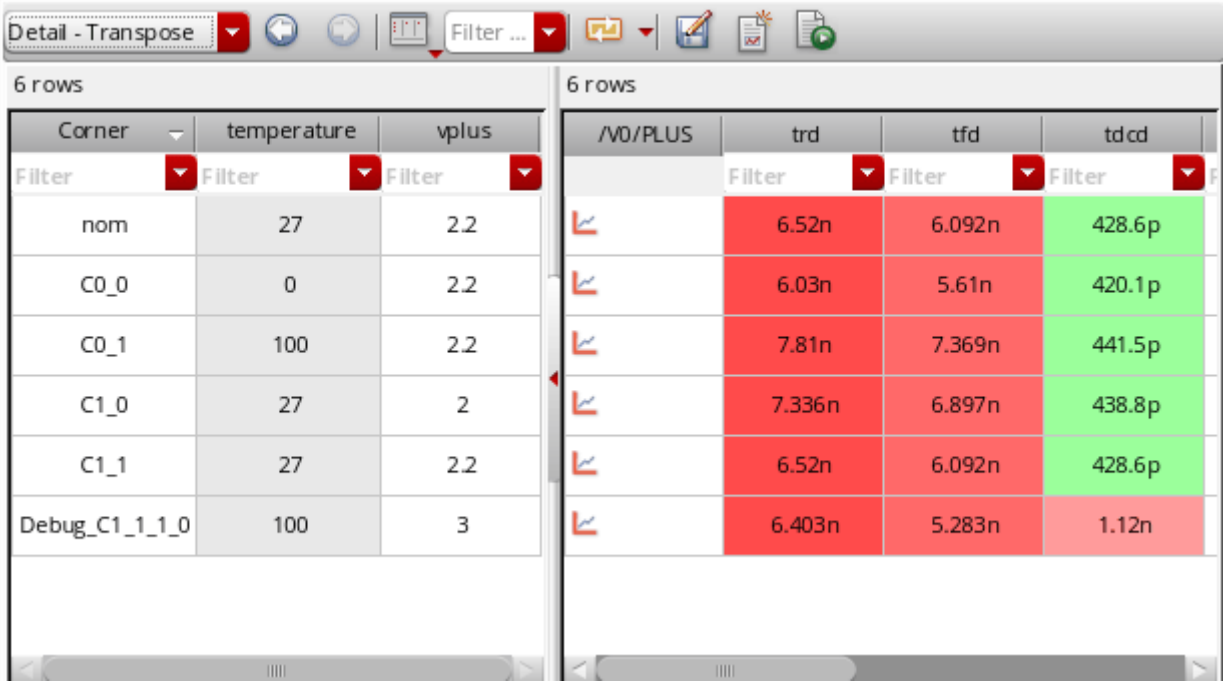



## Detail - Transpose

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


© 2023 Cadence Design Systems, Inc.

simulation point.



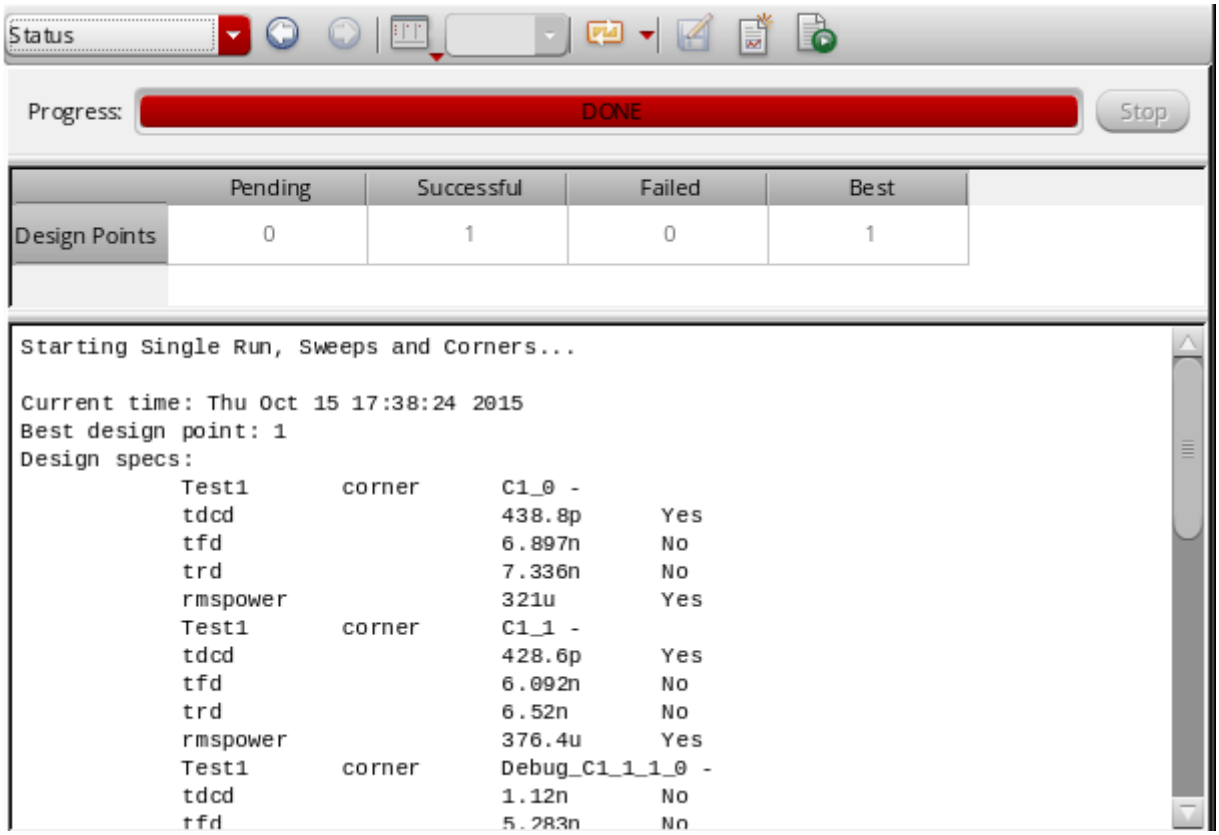

## Optimization

The Optimization view displays the *Test*, *Output*, *Value*, *Spec*, *Weight*, *Min*, *Max* and corners-related columns.


© 2023 Cadence Design Systems, Inc.

## Status

The Status view displays the progress and status of the simulation run, and the contents of the run log file.

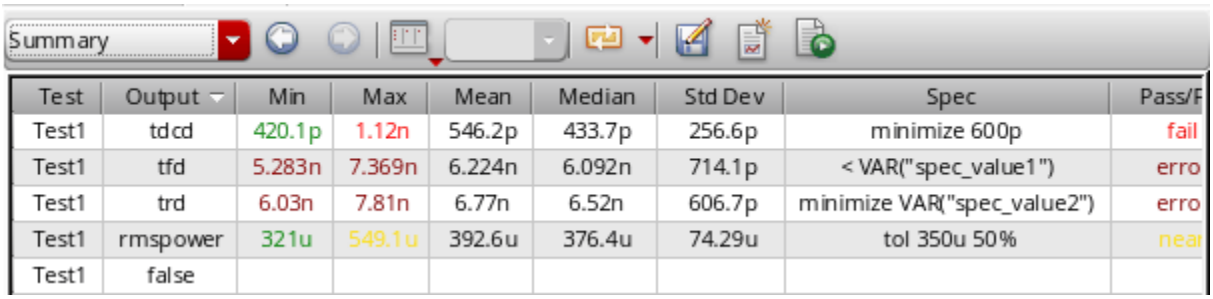




© 2023 Cadence Design Systems, Inc.

## Summary

The Summary view displays the summary of the results for output expressions across all corners and sweeps.




| Test | Output | Min | Max | Mean | Median | Std Dev | Spec | Pass/F |
|---|---|---|---|---|---|---|---|---|
| Test1 | tdcd | 420.1p | 1.12n | 546.2p | 433.7p | 256.6p | minimize 600p | fail |
| Test1 | tfd | 5.283n | 7.369n | 6.224n | 6.092n | 714.1p | < VAR("spec_value1") | erro |
| Test1 | trd | 6.03n | 7.81n | 6.77n | 6.52n | 606.7p | minimize VAR("spec_value2") | erro |
| Test1 | rmspower | 321u | 549.1u | 392.6u | 376.4u | 74.29u | tol 350u 50% | near |
| Test1 | false | | | | | | | |

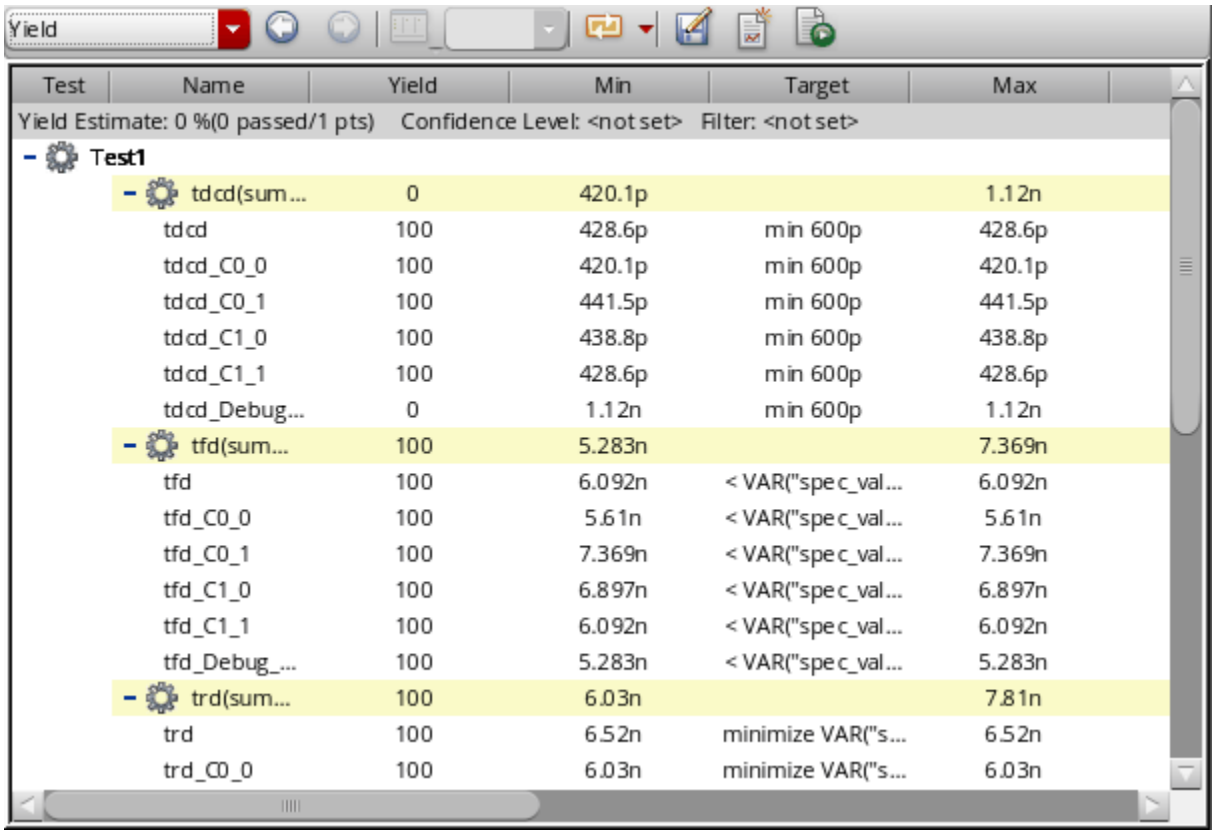
The columns in the Summary view are described below:

| Column | Data/Content |
|---|---|
| *Test* | Name of the test for which the output is defined. |
| *Output* | Name of the output expression. |
| *Min* | Minimum value of the output. |
| *Max* | Maximum value of the output. |
| *Mean* | Mean value of the output. |
| *Median* | Median value of the output. |
| *Std Dev* | Standard deviation of the values for the specification.<br><br>**Note:** For Single Run, Sweeps and Corners, this is the population standard deviation. For other run modes, this is the sample standard deviation. |
| Spec | Specification defined for the output expression. |
| Pass/Fail | The pass/fail status of the specification across all sweeps and corners. |


© 2023 Cadence Design Systems, Inc.

## 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.




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 |
| *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 resulted for an output |


© 2023 Cadence Design Systems, Inc.

| Column | Data/Content |
|---|---|
| *Std Dev* | Sample standard deviation for an output |
| *Sigma To Target* | Distance from the standard deviation value to the target value, defined as follows:<br>`mean value to target value / standard deviation`<br>**Note:** This column is hidden by default. |
| *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, $\mu$ is the estimated mean of the process and $\sigma$ is the estimated variability of the process, Cpk is calculated by using the following equation.<br>$$Cpk = \min\left[\frac{USL-\mu}{3\sigma}, \frac{\mu-LSL}{3\sigma}\right]$$<br>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 |
| *Std Dev* | Worst case value of the standard deviation as computed per corner |
| CV(%s) | Coefficient of Variation or relative standard deviation (in percentage) |
| *Cpk* | Minimum Cpk value across all corners |


© 2023 Cadence Design Systems, Inc.

| Column | Summary value |
|---|---|
| *Errors* | Number of points that resulted in simulation or evaluation errors for the output |

The gray-colored row at the top displays the *Yield Estimate* – an estimate of the circuit yield taking into account all of the specifications.

You can specify a confidence level to view the yield estimate confidence intervals.

*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.

## Capacitance

The Capacitance results view displays the capacitance values saved by Spectre when you enable the allcap parameter in the Save circuit information analysis table of the Save Options form for Spectre. When this parameter is enabled, Spectre runs the netcap analysis as part of its simulation and prints the capacitance contribution from extracted view (DSPF, SPEF), design capacitor, and MOSFET-terminal capacitances.

For more details about this view, see Capacitance in the *Virtuoso ADE Assembler User Guide*.

## 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.

### Hiding and Showing Columns in the Detail - Transpose View

To hide a column, do the following:

➡ Right-click the name of the column and choose *Hide Column*.

To display all the columns, do the following:


© 2023 Cadence Design Systems, Inc.

➡ Right-click 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 a column name.
2. Drag and drop the column at the desired location.

### Viewing Output, Test, and History Information

To view the output, test and history information, 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.

## Using the Status View

See the following topics for more information about using the Status view on the Results tab:

### 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 drop-down on the Results tab. Select the options you want to display. The available options are: *Progress Bar*, *Status Bar*, and *Run Log*. A check mark next to the option indicates that information for that option is being displayed.

## Freezing Columns

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.


© 2023 Cadence Design Systems, Inc.

**Note:** You can freeze columns only in the Detail, Yield, and Optimization views in the Results tab.

To freeze columns, do the following:

- Right-click 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 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.


© 2023 Cadence Design Systems, Inc.

## Sorting Results in the Detail and Detail-Transpose Views

You can sort data in the *Detail* or *Detail-Transpose* result views to view the related data rows together in a the ascending or descending order. For example, you can sort rows by the output name, test name, or in the order of result values in any column.

To sort results based on a specific column in the *Detail* or *Detail-Transpose* result views:

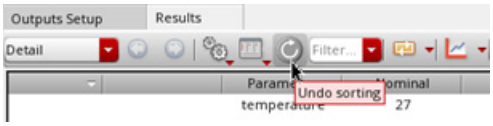
➡ Click the header of the column that you want to use as base for sorting.

All the rows in the results table are sorted according the values in 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.

To undo sorting of results:

➡ Click *Undo sorting* on the toolbar of the *Results* tab.

The results are rearranged in the default order as they were visible before sorting.



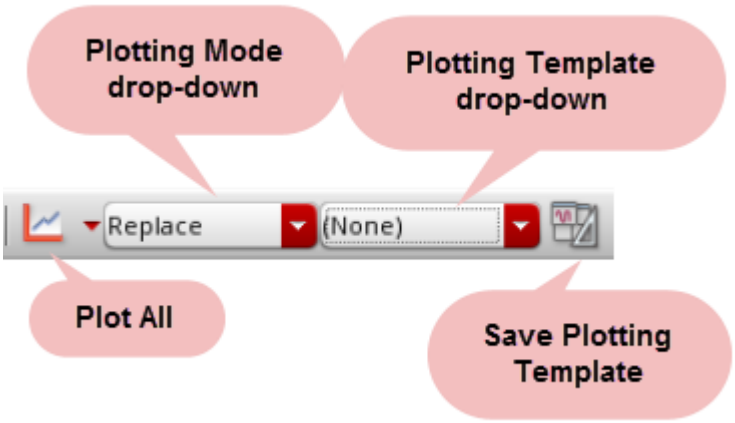


© 2023 Cadence Design Systems, Inc.

# Working with Plotting Templates

You can create a plotting template in ADE Explorer and ADE Assembler to plot resultant waveforms, signals, or scalar values in a specified format in the Virtuoso Visualization and Analysis XL window.

The plotting template drop-down list box is available on the plotting toolbar in the *Results* tab, as shown in the figure below:




**Note:** The plotting template drop-down list box is enabled only when the plotting mode is `Replace` or `New Win`. If the plotting mode is set as `Append` or `New SubWin`, this drop-down is disabled.

The plotting template drop-down list box is available by default. However, you can set the following environment variable in CIW to disable this feature.

```
envSetVal("maestro.gui" "usePlottingTemplate" 'boolean nil)
```

### Creating and Saving a New Plotting Template

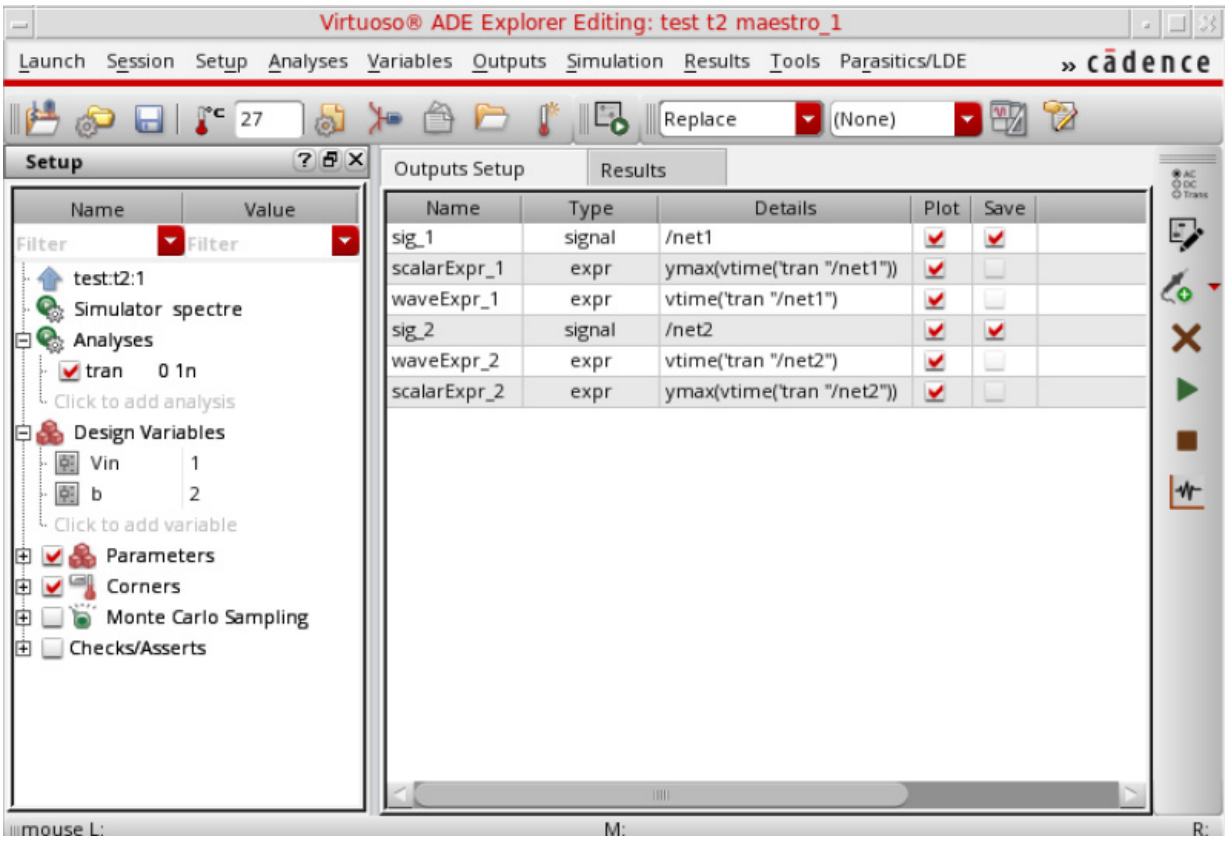
You can create a new plotting template to save the customizations in Virtuoso Visualization and Analysis XL, and reuse this template later to plot waveforms using ADE Explorer and ADE Assembler.

**Note:** The plotting template can be created only when the simulation results are generated using IC6.1.8, ICADVM18.1 or later versions of ADE Explorer or ADE Assembler.

The example below shows how you can create a new plotting template in ADE Explorer.


© 2023 Cadence Design Systems, Inc.

Consider the following ADE Explorer setup with signals, waveform expressions, and scalar expressions added as outputs to be evaluated during the simulation run.

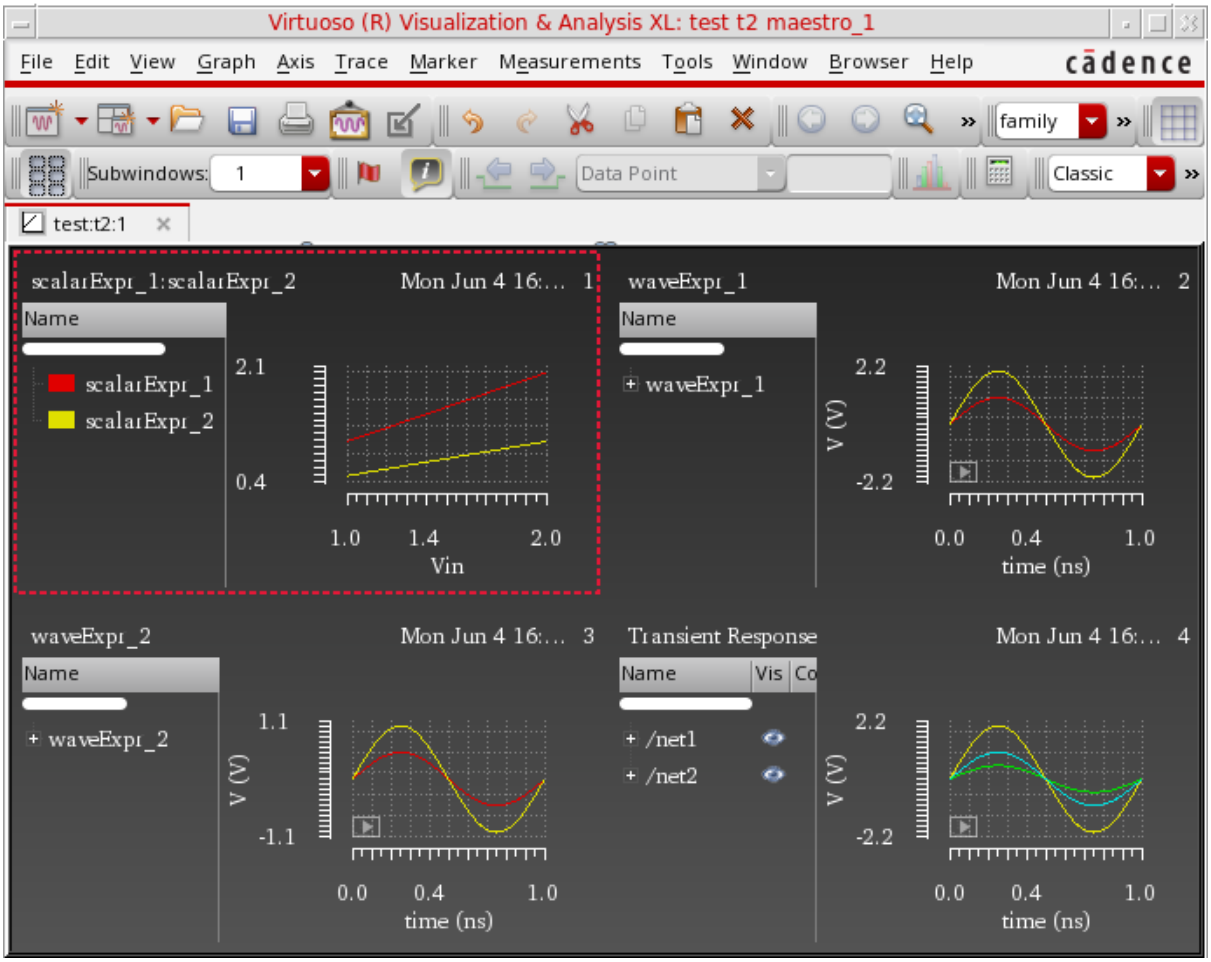



When you run the simulation, the results are displayed in the *Results* tab and waveforms are plotted in the Virtuoso Visualization and Analysis XL window. You can further customize the graph by modifying their color, size, fonts, and layout. You can also zoom into the graphs, split traces across strips, and add markers. For example, the figure below shows the following modifications in the graph window:

- Zoomed into subwindows 2 and 3
- A horizontal marker added in subwindow 1


© 2023 Cadence Design Systems, Inc.

- A vertical marker added in subwindow 2




If you save this graph window as a plotting template, you can reproduce this plot configuration in subsequent simulation runs.

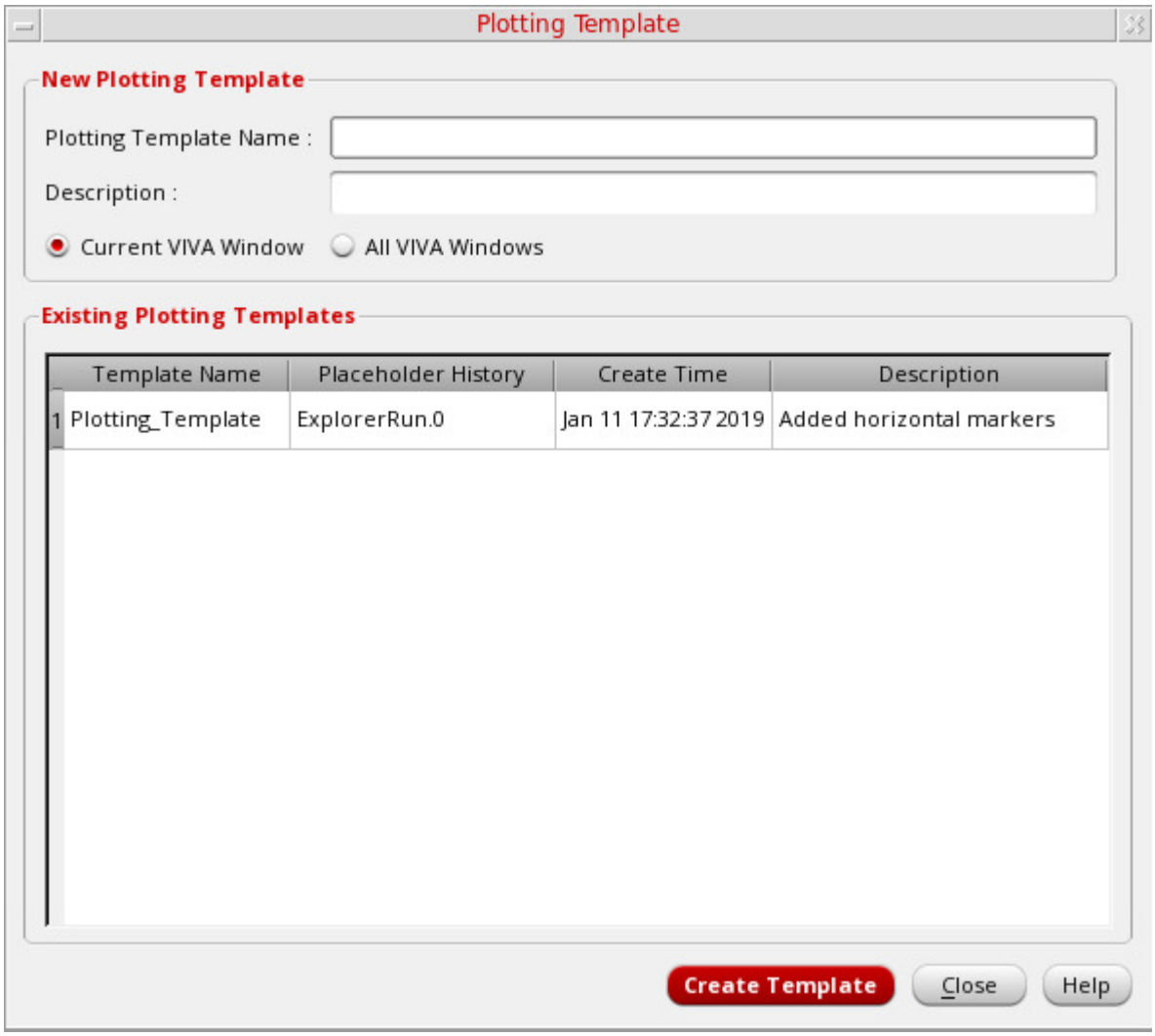
### *Saving the Plotting Template*

You can save a graph window as a plotting template using Virtuoso Visualization and Analysis XL or ADE Explorer. This section explains how this graph can be saved as a plotting template, named `Template2`.

1. In Virtuoso Visualization and Analysis XL window, choose *File – Create Plotting Template*.
2. In ADE Explorer, click *Save Plotting Template* ( ) on the toolbar.


© 2023 Cadence Design Systems, Inc.

The Plotting Template form is displayed, as shown in the figure below:




3. In this form, specify the values of following fields:

   In the *New Plotting Template* section:

   a. In the *Plotting Template Name* field, specify a name that you want to use to save the plotting template.

   b. In the *Description* field, provide a brief description about the plotting template to be saved.

   c. Select the *Current VIVA Window* option to save the current graph window only. This option is selected by default.
   Select the *All VIVA Windows* option to save all the windows opened in Virtuoso Visualization and Analysis XL for different analyses.

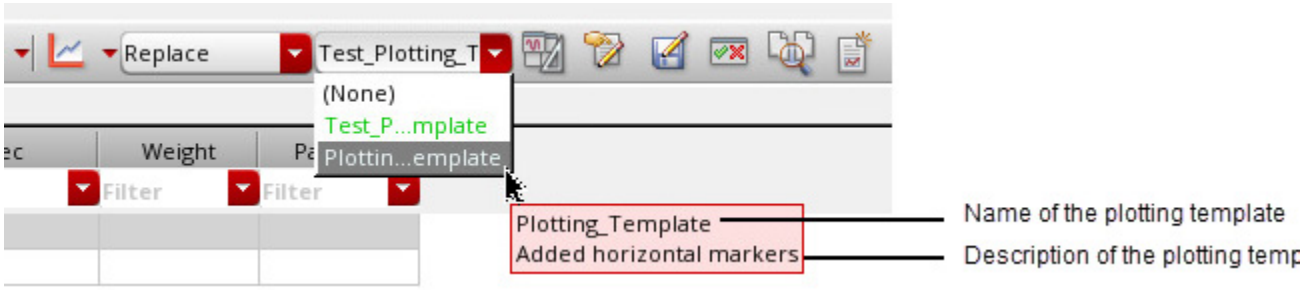

© 2023 Cadence Design Systems, Inc.

**Note:** These options are disabled if no graph window is open for creating the plotting template.

**d.** The *Existing Plotting Templates* section displays all the previously saved plotting templates. You can change the *Template Name* and *Description* fields by double-clicking the text. To delete an existing plotting template, right-click and choose *Delete*.

**e.** Click *Create Template* to create and save the specified plotting template.

The specified plotting template is saved and displayed in the *Plotting Template* drop-down list in ADE Explorer, as shown in the figure below.



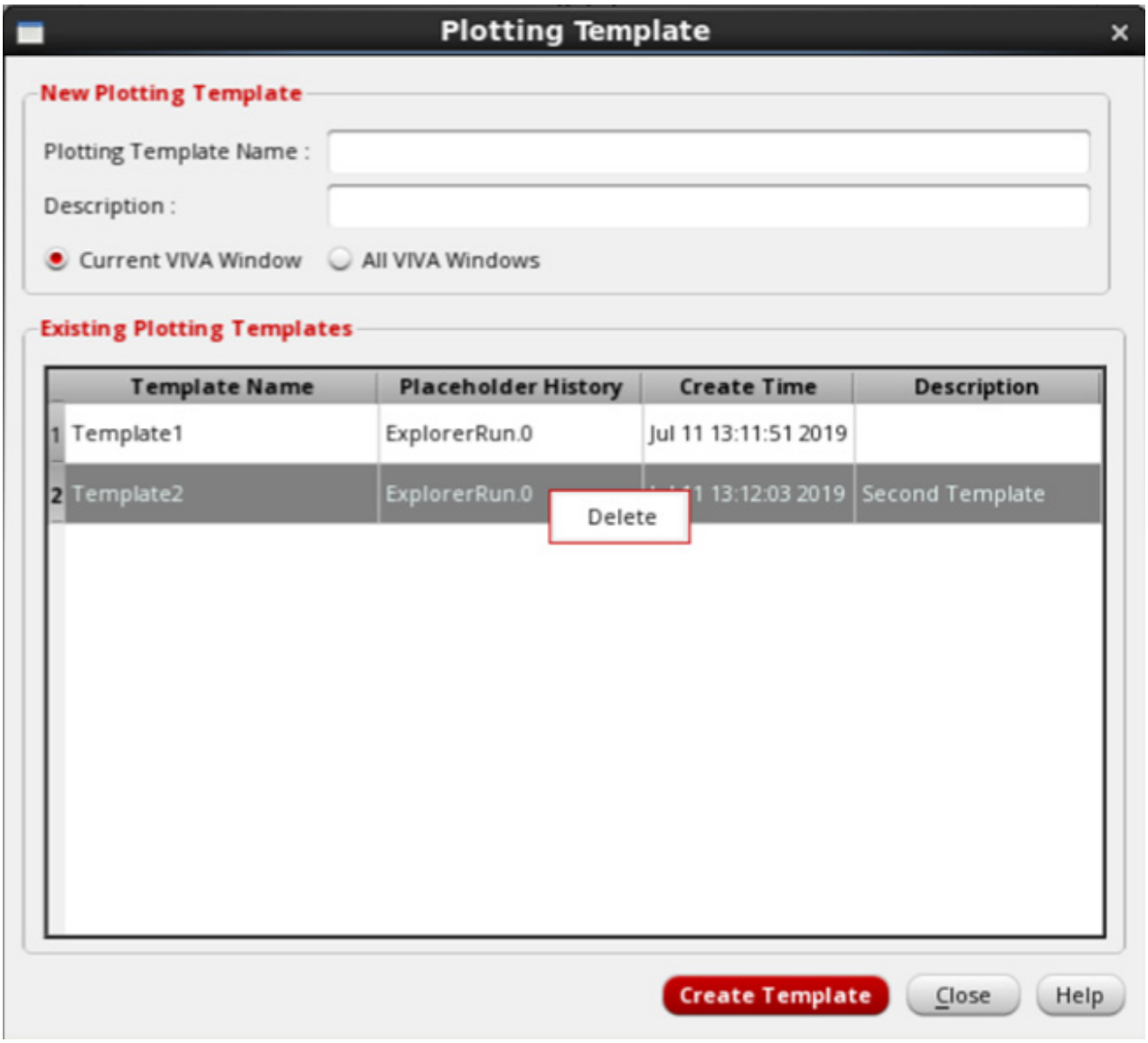

**Note:** When you hover the pointer on a plotting template, name and description of the plotting template are displayed.

You can use this template to plot the waveform outputs in the specified format. All the formatting, markers, layout changes, and other interface changes will be retained as specified in the plotting template. Select the newly created plotting template in the *Plotting Template* drop-down list box before running the simulation.

By default, ADE Explorer plots waveform results automatically when the simulation is complete. However, if the automatic plotting is not selected, you can select the plotting template any time after the simulation results are generated and then click *Plot All* to apply the template. This is the default setting in ADE Assembler.


© 2023 Cadence Design Systems, Inc.

To delete a plotting template, right-click the template in *Existing Plotting Templates* section and select *Delete*.




### Important Points to Note

- If you do not want to use a plotting template, select *None* from the Plotting Template drop-down list box. The waveforms are then plotted in the default plotting format.
- The plotting template does not support the following context-sensitive plotting options that appear when you right-click a result on the *Results* tab: *Plot*, *Plot across corners*,


© 2023 Cadence Design Systems, Inc.

*Plot All*, and *Plot across design points*. When any of these options are selected, the waveforms are plotted in the default plotting format.

- You can share the plotting template across multiple `maestro` cellviews. To save the plotting template in a cellview, use the Save A copy form that appears when you choose *Session – Save A Copy*. You can import an existing plotting template using the Import Setup form that appears when you choose *Session – Import*.
- The plotting template created in ADE Assembler can be reused in ADE Explorer and vice-versa. If a template is created from more than one tests in ADE Assembler, other tests are ignored while plotting with template in ADE Explorer. For history `Interactive.X` in ADE Assembler, the corresponding history name in ADE Explorer is `ExplorerRun.0`.
- The waveform results generated from the direct plot run cannot be used to create a plotting template in ADE Explorer. In this case, to create a template, plot the waveform results in Virtuoso Visualization and Analysis XL and then create the plotting template.
- The Create Datasheet form also supports the plotting of waveforms using plotting templates. If you select a plotting template and select the *Plot All* option, the output waveforms are plotted in Virtuoso Visualization and Analysis XL. Now, if you create a datasheet using the Create Datasheet form with the Open Graphs option selected, the output waveforms in the datasheet are plotted based on the specified plotting template.
- For multi-point simulation runs in ADE Assembler, if you plot only one corner or sweep point and create the plotting template, then when you do *Plot All* and apply this template, only the corner specified in the template is plotted.
- In Assembler, if you plot the template using the same interactive session as the placeholder history, only the signals that match the same corner and design variable are plotted.

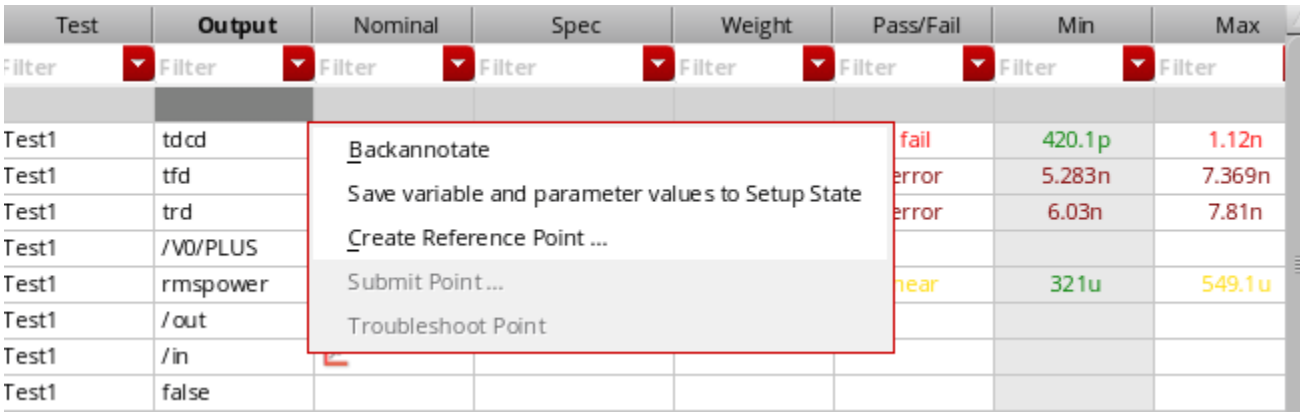

© 2023 Cadence Design Systems, Inc.

# Backannotating from ADE Explorer Results

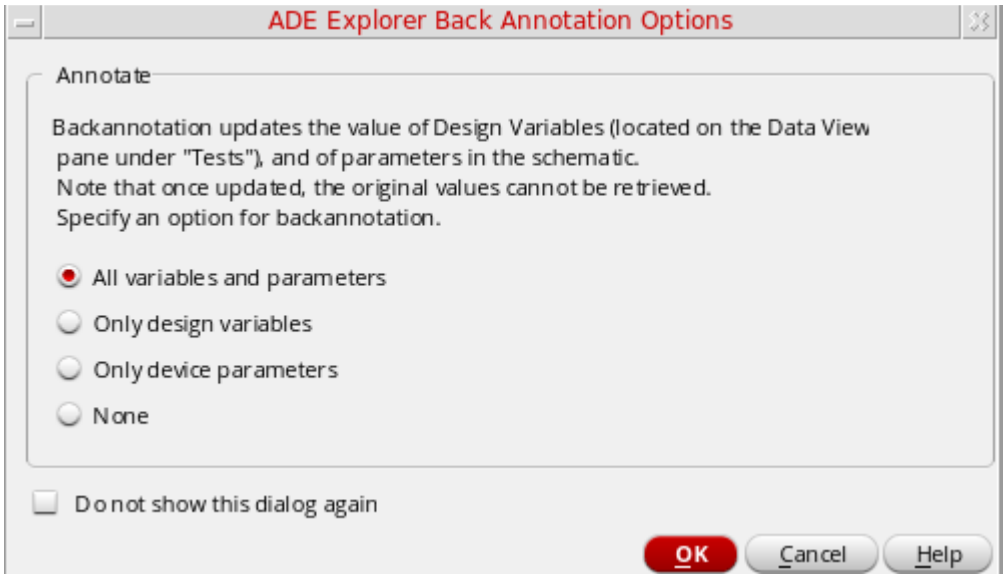
After running an ADE Explorer simulation, you can backannotate the results from the best or the desired design point to make appropriate changes in the design and simulation setup. When you backannotate the results from a design point, the values of design variables used for that point are copied to the *Design Variables* list in the Setup assistant of the ADE Explorer setup and the values of the device parameters are copied to the schematic view of the design.

To backannotate the ADE Explorer results, do the following:

1. Ensure that the Details view is open in the Results view tab.
2. Right-click the gray bar on top of the selected design point and choose *Backannotate*, as shown in the figure below:




The ADE Explorer Back Annotation Options form is displayed, as shown below.





© 2023 Cadence Design Systems, Inc.

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 Explorer 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 Explorer setup.

**Note:** Backannotate option is not available in ADE Explorer for a single point simulation run.


© 2023 Cadence Design Systems, Inc.

# Re-evaluating Expressions and Specifications

After running a simulation, if you add, modify or delete output expressions or specifications on the Outputs Setup tab, you can re-evaluate the expressions and specifications without having to rerun the simulation. When output expressions or specifications are re-evaluated, the updated results are displayed in the Results tab. The results database is also automatically updated.

To re-evaluate expressions and specifications for the last run history, do one of the following:

- Click drop-down on the Results tab to re-evaluate both expressions and specifications. The following menu is displayed:

- 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.
  - Select *Partial Data* to re-evaluate both expressions and specifications using the partially complete simulation data. You can use this option to evaluate the partial results if the simulation run could not complete due to some reason but there are significant results that can be used to analyze the results. For any output value that is evaluated on the partial data, the tooltip shows the text *[Evaluated on partial data]*.

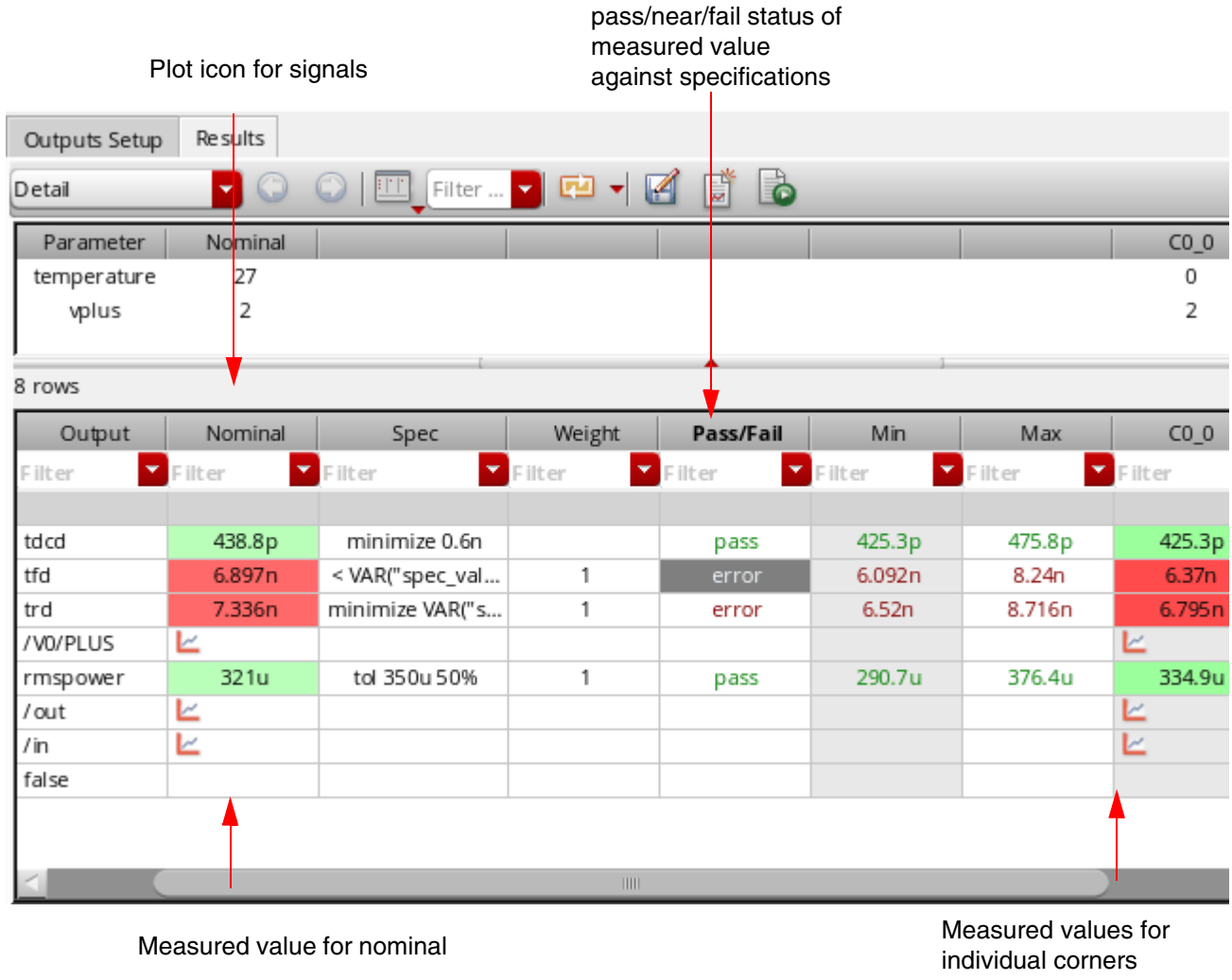
### *Important Points to Note*

- By default, re-evaluation is done for incremental changes. That is, only the results for the revised outputs are evaluated again. However, you can forcefully re-evaluate all the results by selecting the all check boxes in the drop-down menu of the command.
- When you re-evaluate expressions and specifications, the updated results will not be plotted.


© 2023 Cadence Design Systems, Inc.

# Viewing Specification Results in the Results Tab

When you run a simulation, the simulation results are displayed in the Results tab. For more information about running simulations, see Chapter 7, “Running a Simulation”.




*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.


© 2023 Cadence Design Systems, Inc.

The results will also be updated based on the new or modified expressions and specifications if you click this button on the Results tab to plot across all points.

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.
- If a specification is specified as an expression, OCEAN script, or SKILL code, the evaluated value of the expression or code is compared with the output value to identify the *pass* or *fail* status. To view the evaluated specification value, move the mouse pointer over the *Spec* column on the *Results* tab. The evaluated value is displayed in a tooltip.
- *pass*, *near*, *fail,* or *error* 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.
  - *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.
  - *error* means that the expression given in the Spec column for the output failed to evaluate.

    In this case, the result value is also displayed with a bright red background To view the error details, move the mouse over the result value in the corner column. The tooltip displays the details of the error, as shown in the figure given below.

© 2023 Cadence Design Systems, Inc.

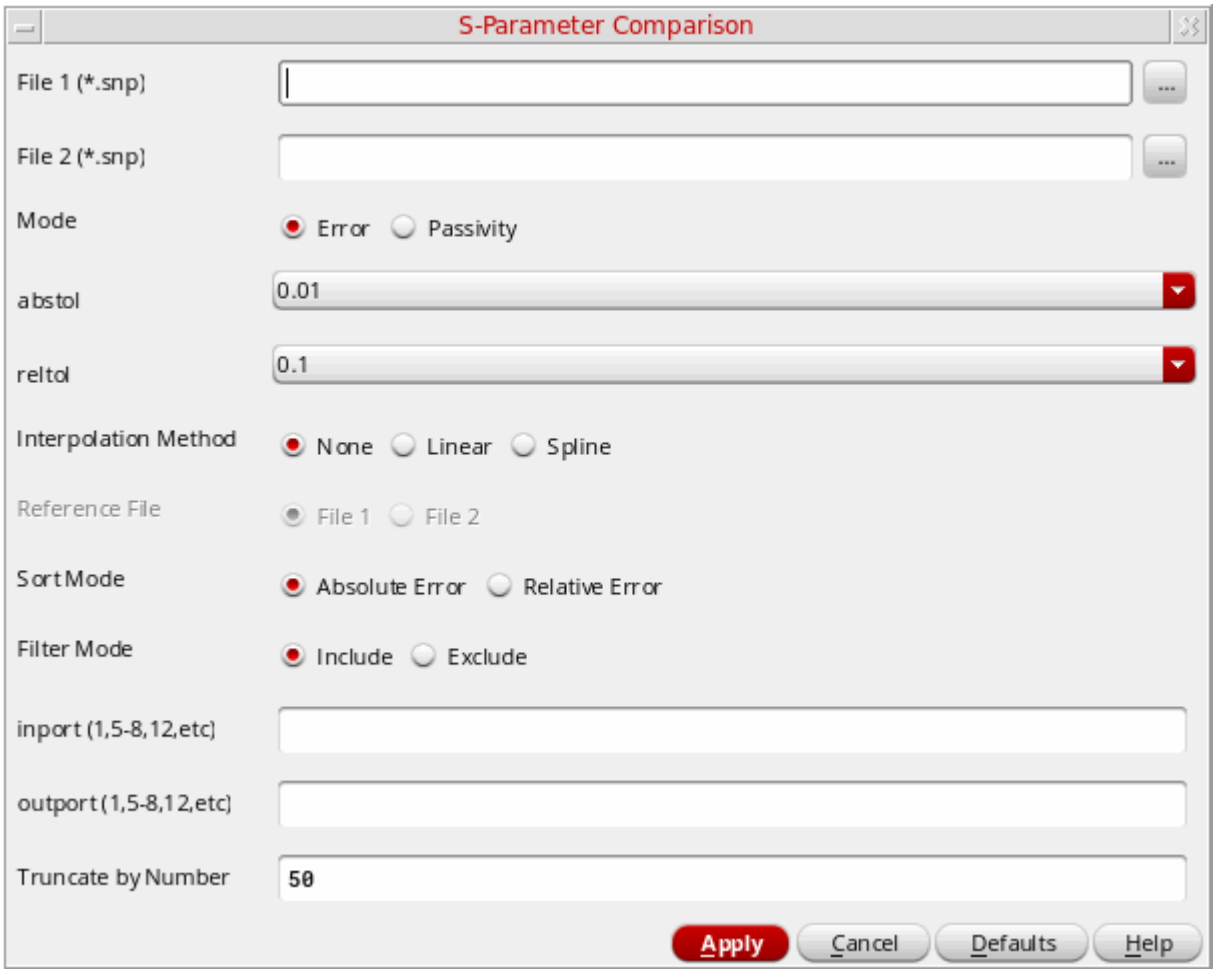
# Comparing S-Parameter Data

You can use the S-parameter comparison form to inspect large S-parameter data for any defects. This data can be large, containing hundreds of frequency points and millions of data points. With the help of this form, you can vary tolerances and perform basic data filtering to zoom in to problem areas. The comparison results are displayed in either error or passivity modes.

To compare the S-parameter data:

➡ Choose *Tools – S-Parameter Comparison*.

The S-Parameter Comparison form appears as shown in the figure below.




In this form, specify the following fields:


© 2023 Cadence Design Systems, Inc.

1. In the *File 1(*.snp)* field, specify the first S-parameter data file used for the comparison. Click the ellipses button to select a file. Ensure that the specified file is in touchstone format with `.snp` extension.
2. In the *File 2(*.snp)* field, specify the second S-parameter data file used for the comparison. Click the ellipses button to select a file. Ensure that specified the file is in touchstone format with `.snp` extension.

**Note:** Both these files should have the same port number and the number of signals in both these files should be equal.

3. In the *Mode* field, select a mode in which you want to generate and display the comparison results. The modes can be `Error` and `Passivity`.
   - Select `Error` if you want to display absolute and relative errors for the specified signal in the comparison results. This is the default mode. When you select this mode, ensure that the specified files have the same port number.
   - Select the `Passivity` if you want to display frequency and passivity violations. In this mode, the results can be generated even if you specify only one S-parameter file. In case you specify both the files, they need not to have the same port number.
4. In the *abstol* field, select the absolute tolerance value. The valid values are `0.1`, `0.01`, `1e-3`, `1e-4`, and `1e-5`. The default value is `0.01`. This field is displayed for both the modes.

   The following fields are displayed only when the mode is selected as `Error`.
   - In the *retol* field, select the relative tolerance value. The valid values are `0.1`, `0.05`, `1e-2`, `1e-3`, and `1e-4`. The default value is `0.1`.
   - In the *Interpolation Method* field, select whether you want to interpolate data when the frequency points in the two S-parameter files are not aligned. The available options are: `None`, `Linear`, `Spline`. The default value is `None`. For more information, see Interpolating S-Parameter Data on page 929.
   - In the *Sort Mode* field, select `Absolute Error` if you want to sort the results by absolute errors and select `Relative Error` to sort the results by relative errors. By default, the `Absolute Error` option is selected.
   - In the *Filter Mode* field, select `Include` if you want to include all the signals and select `Exclude` to exclude all the signals.
   - In the *inport (1,5-8,12,etc)* field, specify the input port number. The different port numbers should be separated by a comma or hyphen and specified in the `1,5-8,12` format. When the specified inport numbers are bigger than the given port number, an error message is displayed.


© 2023 Cadence Design Systems, Inc.

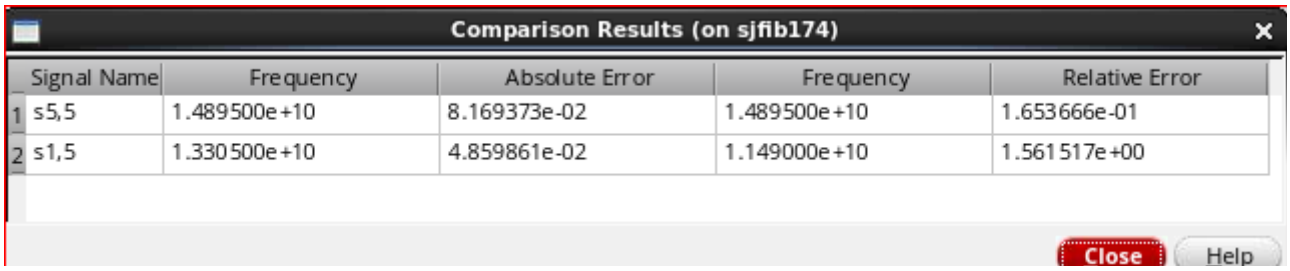
- In the *outport (1,5-8,12,etc)* field, specify the output port number. The different port numbers should be separated by a comma or hyphen and specified in the `1, 5-8,12` format.
- In the *Truncate by Number* field, specify a truncate number used to display the comparison results. The default value is `50`.

5. (Optional) Click *Default* to set the form fields back to their default values.
6. Click *Apply* to start the comparison and generate results. According to the selected mode, the results are displayed in different windows, as explained in the sections below.

## Viewing Results in Error Mode

If you selected the `Error` mode in the S-Parameter Comparison form, the results are listed in the *Comparison Results* table based on the combination of the specified inport and outport values. For example, if the inport value is `1,5` and the outport value is `5`, the signals `s5,5` and `s1,5` are displayed in the comparison results table. Each row in the results table displays the problematic frequency for absolute error and relative error calculated for the given port pair.

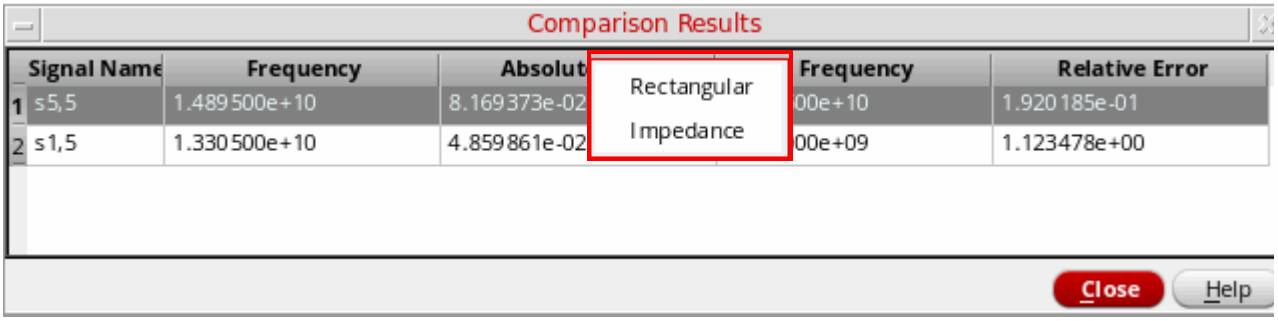
The number of comparison results are displayed according to the value specified in the *Truncate by Number* field and the results are sorted in the decreasing order of the absolute or relative error values.



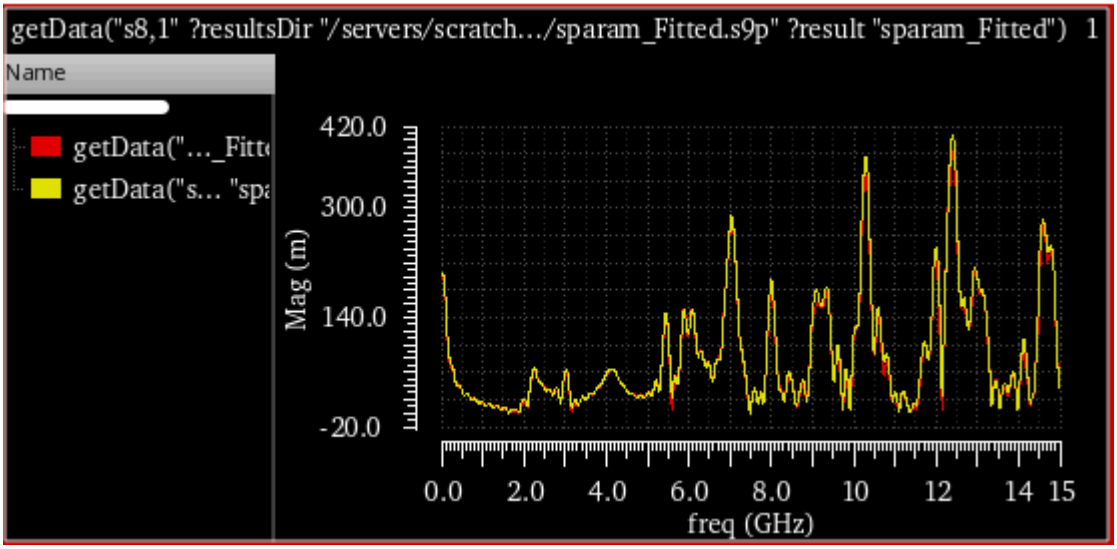


© 2023 Cadence Design Systems, Inc.

To plot a specified signal displayed in the comparison results in a rectangular or circular (impedance) graph, right-click the signal row and choose *Rectangular* or *Impedance*.

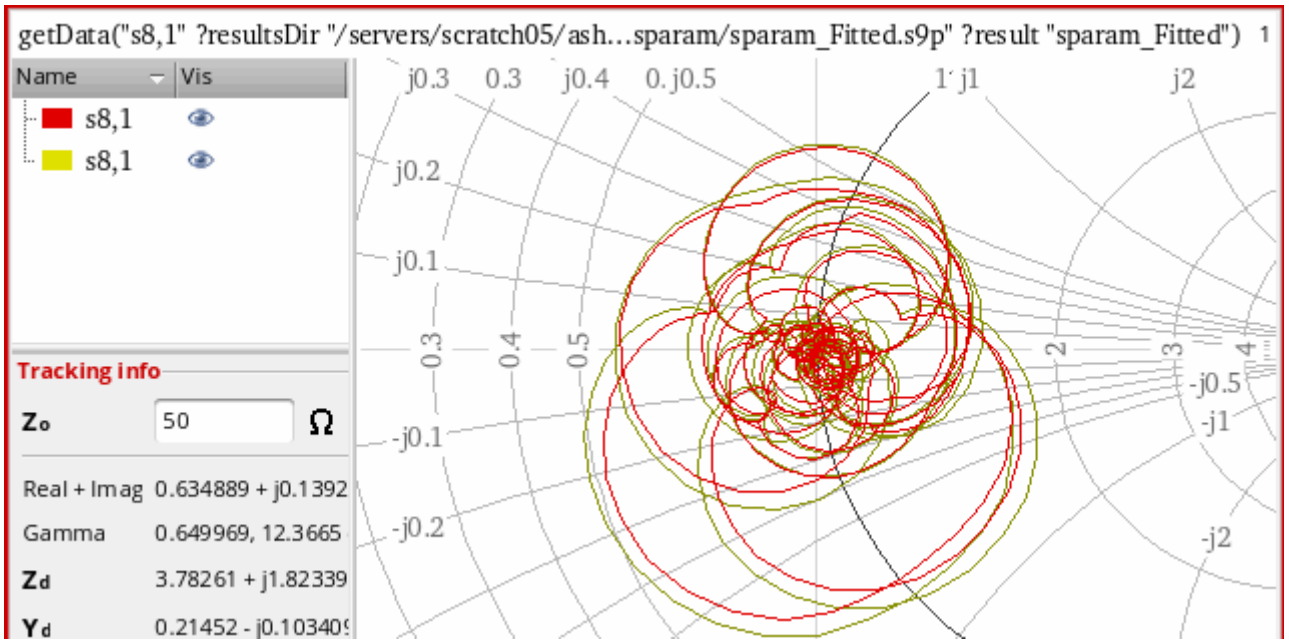



When you choose the *Rectangular* option, the following waveform showing the magnitude of the selected signal is plotted in the graph window:



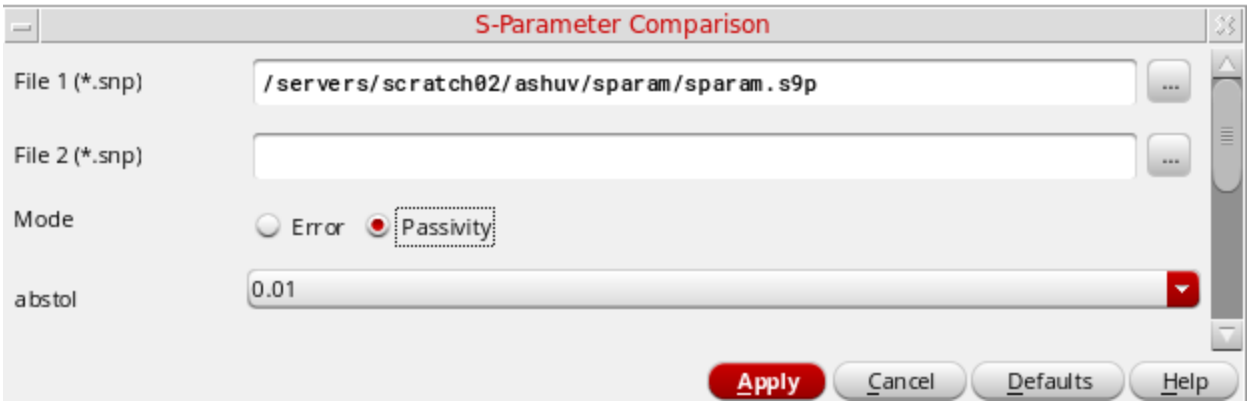

The following waveform is displayed when you choose the *Impedance* plot option:


© 2023 Cadence Design Systems, Inc.




## Viewing Results for Passivity Mode

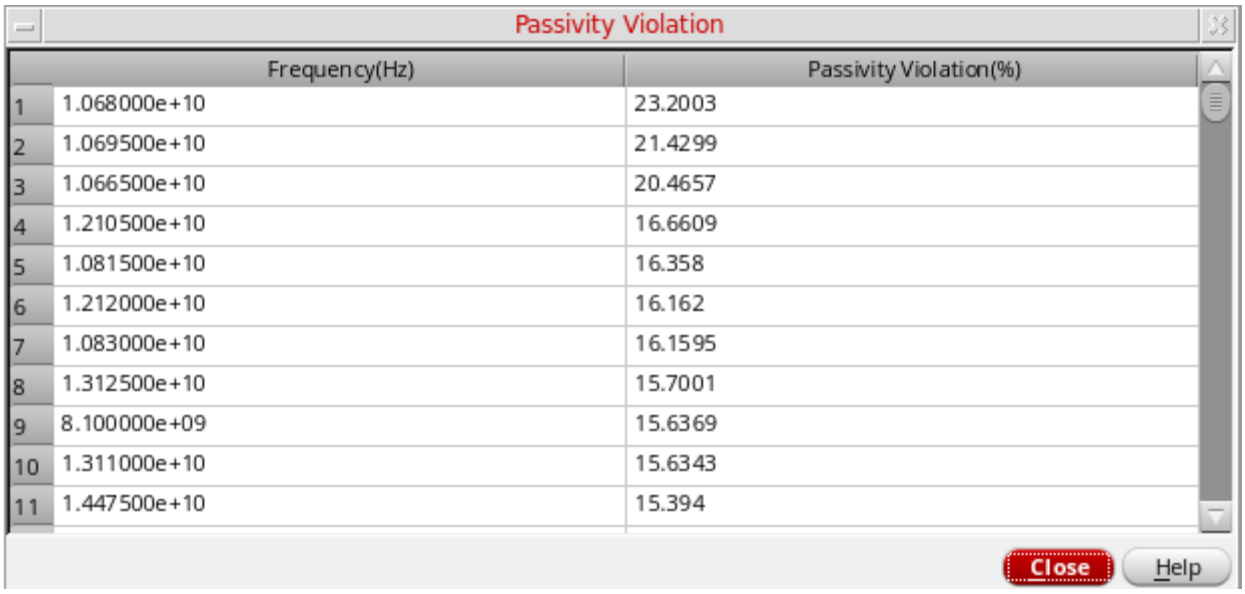
The passivity violation results are displayed if you selected the `Passivity` mode and specified the absolute tolerance in the S-Parameter Comparison form, as shown in the figure below. The passivity violation results can be generated even if you specify only one S-parameter file. In case you specify both the files, they need not to have the same port number.



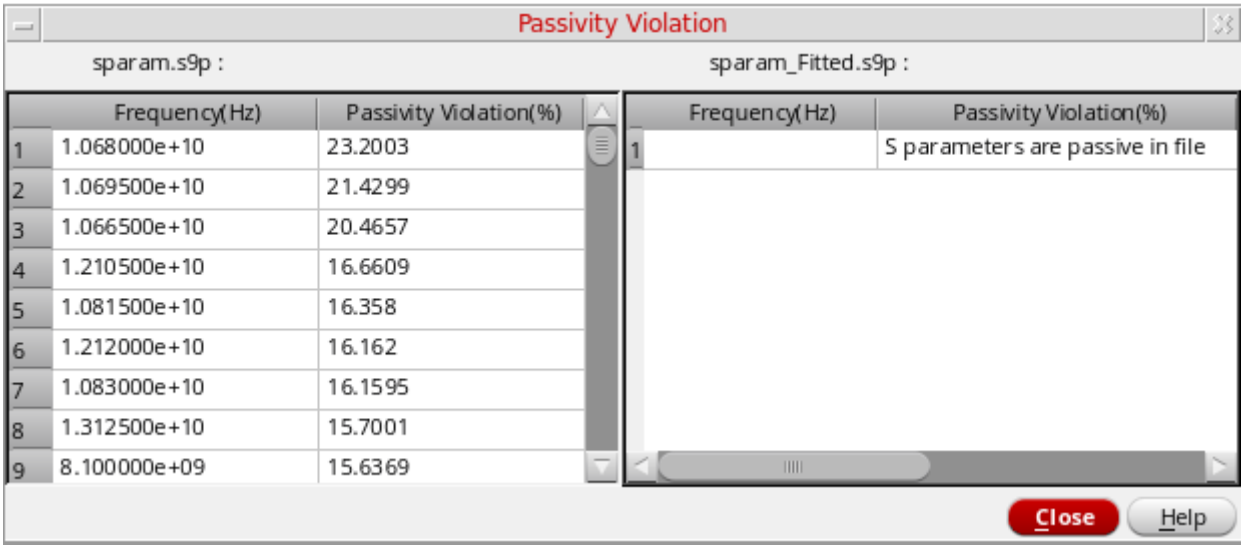


© 2023 Cadence Design Systems, Inc.

The following figure shows the results when you specify only one file. The Passivity Violation results display a list of frequency and passivity violation (in %) of the specified S-parameter files.




The following figure shows the results generated when you specify both the files. The file, `sparam_Fitted.s9p`, does not list any violations because all the S-parameters in are passive in this file.





© 2023 Cadence Design Systems, Inc.

## Interpolating S-Parameter Data

The S-parameter interpolation can be performed by using the following two ways:

- Interpolation in `nport`, which is set by the option `interp=bbspice/linear/spline` and done by the simulator.
- Interpolation using the S-parameter Comparison form. This is done by selecting the interpolation method as `Linear` or `Spline` and then selecting the reference file.

By default, the *None* option is selected, which means no interpolation is performed. In this case, the *Reference File* option is not available. The `None` option should be used for files that have identical frequency points and when `nport interp=bbspice`.

*Caution*

***The interpolation methods listed above should be used independently. Based on your application, you can use either bbspice or linear or spline in nport simulations. The interpolation in the S-parameter Comparison form is only for S-parameter data comparison purpose and does not make any assumptions of the interpolation method adopted in the simulation.***

### Performing Linear or Spline Interpolation

When the frequency points in the two S-parameter files are not aligned, you can interpolate the S-parameter data by using the linear or spline interpolation methods. These options should be used when `nport interp=linear/spline`. To perform the linear or spline interpolation,

1. Choose *Interpolation Method* as `Linear` or `Spline`.
2. Choose *Reference File* as either File 1 or File 2. Notice that File 1 and File 2 are already specified in the S-Parameter Comparison form.

   Based on the specified file, linear or spline interpolation is performed on the other file so that the two files have the same frequency points, according to which the S-parameter comparison is to be done.

The following cases can occur:


© 2023 Cadence Design Systems, Inc.

- Case 1: If reference file is File 1, and its frequency range is shorter than that of File 2.

  In this case when linear or spline interpolation method is specified, first the points in File 2 are interpolated and then comparison of File 2 is performed against File 1.

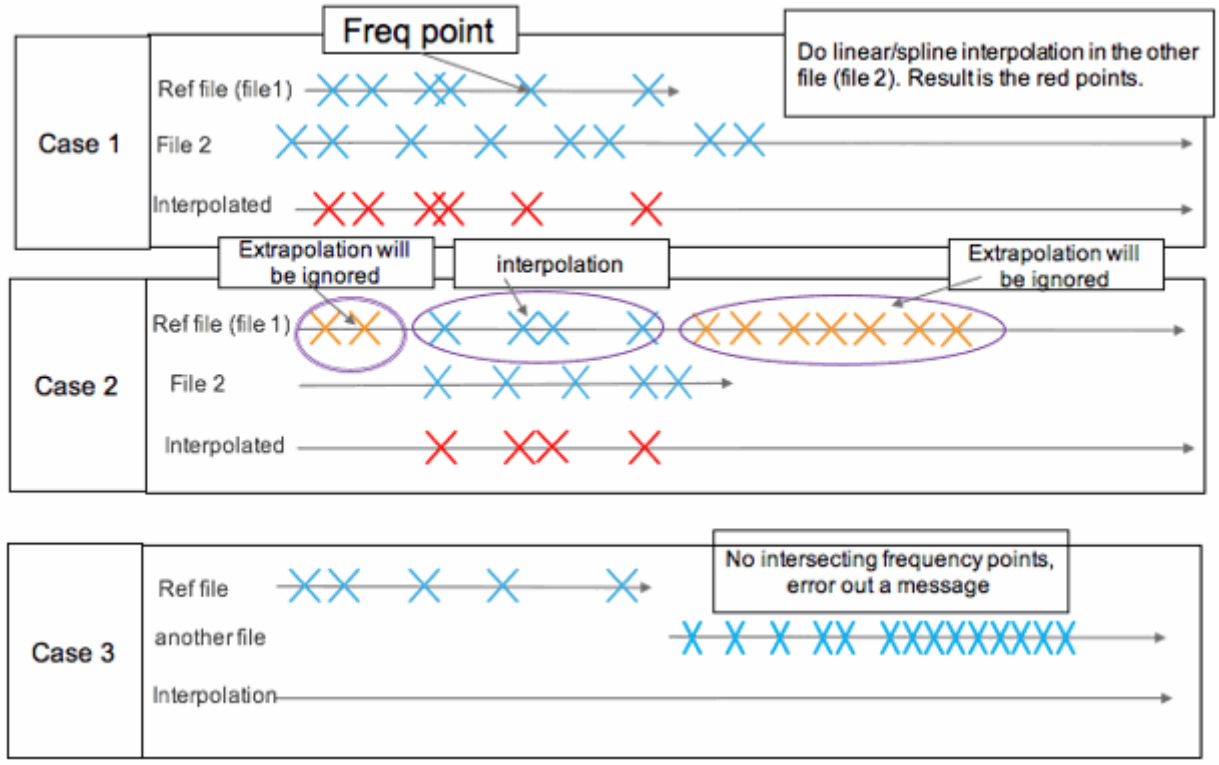
- Case 2: If reference file is File1, and its frequency range is longer than that of File 2.

  In this case, theoretically both interpolation and extrapolation should occur. However, the S-parameter Comparison tool ignores extrapolation and displays a warning message. Then, only interpolation is performed and the comparison is done.

- Case 3: If File 1 and File 2 do not have intersections.

  In this case, no interpolation is performed and an error message is displayed.

The figure below explains these cases graphically:





© 2023 Cadence Design Systems, Inc.

# Working with Datasheets

- Creating a Datasheet for a Checkpoint on page 931
- Displaying Customized Waveform Images in the Datasheet on page 940
- Opening a Datasheet on page 940
- Customizing the Datasheet Format and Structure on page 941

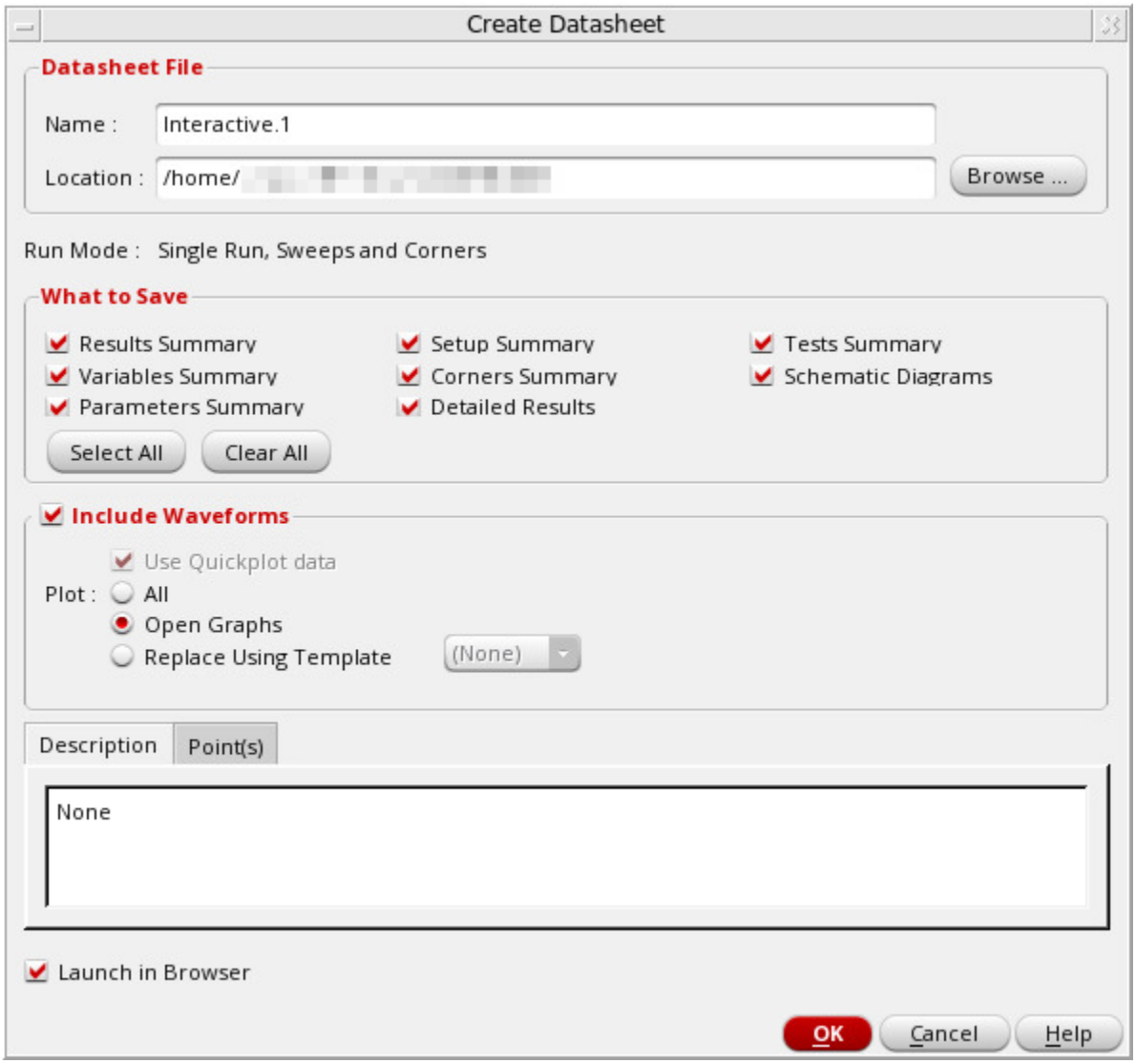
## Creating a Datasheet for a Checkpoint

To create a datasheet for a checkpoint, do the following:

1. Choose *Tools – Create Datasheet*.


© 2023 Cadence Design Systems, Inc.

The Create Datasheet form appears.




**Note:** You cannot create a datasheet for a check point when simulation is in progress.

2. In the *Name* field, type a name for the datasheet.

   **Note:** The default name of the datasheet will be `ExplorerRun.0`.

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.


© 2023 Cadence Design Systems, Inc.

**c.** Click *Open*.

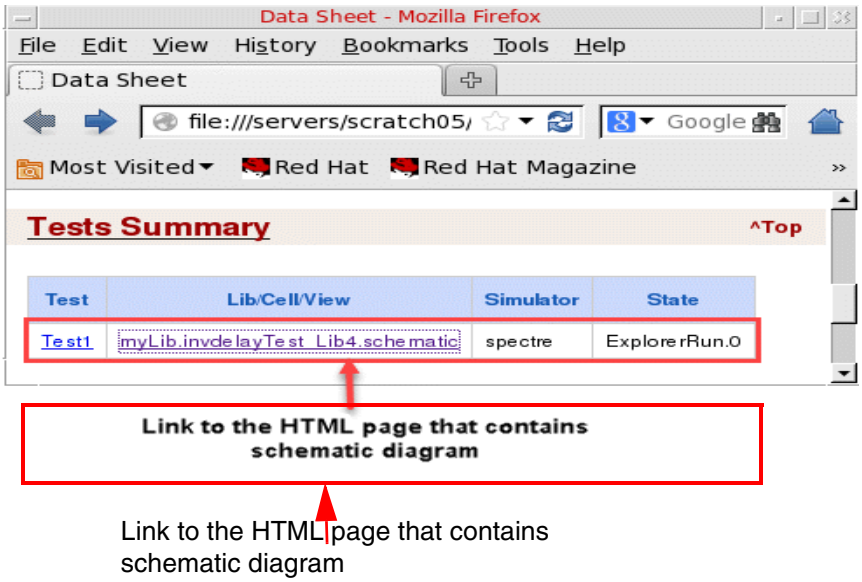
Note the following:

- By default, the datasheet is created in the `documents` directory of the ADE Explorer 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 |
|---|---|
| *Variables Summary* | When turned on, this option writes variables summary information to the datasheet, including names and values of variables.<br><br>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.<br><br>Corners summary information is displayed in the *Corners* section in the main datasheet page. |

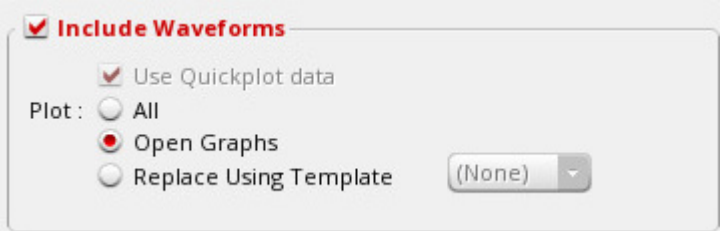

© 2023 Cadence Design Systems, Inc.

| Check Box | Description |
|---|---|
| Schematic Diagrams | When selected, this option saves the schematic diagrams in the datasheet. These diagrams are not displayed on the main datasheet page. Instead, a link to the diagram is added to the *Tests Summary* section. When you click the hyperlink for the saved schematic diagram, a `.html` page saved by Virtuoso Schematic Editor is displayed. Diagrams of the schematic of the design and the devices instantiated in it are displayed on this page. The top of the page shows a Table of Contents (TOC) and a graphical TOC. Click the links in the TOC to view a particular schematic diagram.<br>  <br><br>Link to the HTML page that contains schematic diagram |
| *Tests Summary* | When turned on, this option writes test summary information to the datasheet, including test name, test design, simulator, and state information.<br><br>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.<br><br>Parameters summary information is displayed in the *Parameters* section in the main datasheet page. |


© 2023 Cadence Design Systems, Inc.

| Check Box | Description |
|---|---|
| *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.<br><br>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 939. |
| Setup Summary | Displays the summary of setup used to run the simulation.<br><br>Setup summary information is displayed in the *Tests Summary* section in the main datasheet page. |
| *Launch in browser* | When turned on, this option launches the datasheet in an web browser.<br><br>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`.<br><br>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 Studio Design Environment:<br><br>`ui webBrowser string "browserName"`<br><br>where `browserName` is the name of the browser's executable. The default value for this environment variable is `netscape`.<br><br>**Note:** Ensure that your web browser supports XML and XSLT. |

5. Select the *Include Waveforms* check box to display waveform images in the datasheet files. This check box is selected by default.





© 2023 Cadence Design Systems, Inc.

- Select the *Use Quickplot data* check box if you want to use the quick plot data to print waveforms in the datasheet. This option helps quickly print waveforms in the datasheet against regular plot. When this check box is selected, quick plot is used for the transient data and regular plot is used for the non-transient data. The default setting of this check box is determined by the value of the useQPDataToCreateDataSheet environment variable.
- In the *Plot* field,
  - Select *All* if you want to plot all the graphs in the Virtuoso Visualization and Analysis XL graph window and include them in the datasheet. The *Use Quickplot data* check box field is available only when you select the *All* option.
  - Select *Open Graphs* to include graphs that are currently open in the graph window. When this option is selected, the *Use Quickplot data* check box field is not available.
  - Select *Replace Using Template* if you want to plot the graphs from a plotting template, which you saved earlier, and include them in the datasheet. You can then select the plotting template from the drop-down list displayed next to it. When this option is selected, the *Use Quickplot data* check box field is not available.

  Note the following:

  - If you select the *Open Graphs* option, but no graph is plotted in the graph window, then the plot all command is run to plot all the graphs. These graphs are then printed in the datasheet.
  - If you select the *All* option, but some waveforms are already plotted in the graph window, then new waveforms are plotted based on the mode you have selected in the *Plotting Mode* drop-down list on the *Results* tab. The available plotting modes are: append, replace, new window, and new subwindow.
  - If you select a plotting template to plot the output waveforms, then the waveforms in the datasheet are also plotted based on the selected plotting template.

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 938.

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 Datasheet on page 940.


© 2023 Cadence Design Systems, Inc.

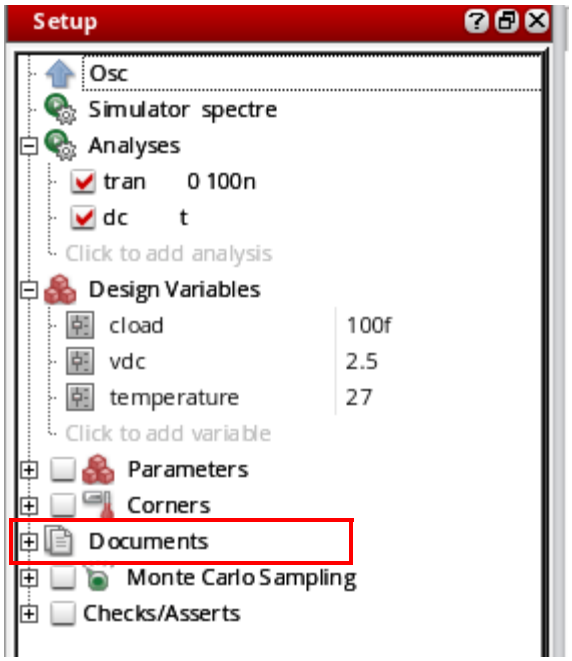
6. (Optional) In the *Points* tab, do one of the following:

| Select | To |
|---|---|
| *Design Point ID* | Include results information only for a specific design point in the datasheet.<br><br>Type the ID of the design point in the *Design Point ID* field.<br><br>**Note:** You can find the ID for a design point from the *Point* column in the Results tab. |
| *All Design Points* | Include results information for all the design points in the datasheet. |

7. (Optional) In the *Description* tab, enter a description for the datasheet.
8. 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.

When you create a datasheet after the simulation run is complete, the Setup assistant starts displaying the Documents section.





© 2023 Cadence Design Systems, Inc.

### 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:

| File Name | Description |
|---|---|
| *<datasheet_name>*.xml | The main datasheet file.<br><br>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.<br><br>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:<br><br>■ `myDatasheet_AC1.xml`<br><br>■ `myDatasheet_TRAN1.xml` |
| dsImg*.png | Portable Network Graphic (`.png`) files for the waveform images displayed in datasheets. |


© 2023 Cadence Design Systems, Inc.

### 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. |


© 2023 Cadence Design Systems, Inc.

| Cell Background Color | Description |
|---|---|
| Red<br>(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. |
| Gray<br>(in *Target* column) | Indicates that a corner specification is disabled. |

## Displaying Customized Waveform Images in the Datasheet

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, click the button.

   Waveforms are displayed in waveform windows.

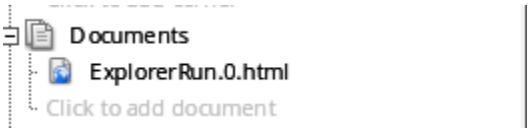
2. Customize the waveforms as required in the waveform window.
3. Create the datasheet without closing the waveform windows in which you customized the waveforms.

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 Setup assistant, double-click 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`.

- Open the `.xml` files in the datasheet directory in a web browser.


© 2023 Cadence Design Systems, Inc.

## Customizing the Datasheet Format and Structure

The following stylesheet files are used to control the format and structure of datasheets.

| 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. |


© 2023 Cadence Design Systems, Inc.

© 2023 Cadence Design Systems, Inc.

# 12

# Performing Reliability Analysis

This chapter describes how to perform RelIability analysis, using the Cadence® Virtuoso® Reliability Simulator interface in ADE Explorer. This interface is only supported for Spectre and AMS.

**Note:** The Reliability Simulator interface in ADE Explorer is supported with the Spectre Simulator only if you are using MMSIM 7.2 or a later version.

See the following topics for more information:


- Introduction on page 944
- Specifying Reliability Options on page 944
- Running the Reliability Simulation on page 956
- Viewing the Reliability Simulation Results on page 957
- Viewing the Reliability Aged Netlist on page 959
- Annotating Simulation Results to the Schematic on page 959



© 2023 Cadence Design Systems, Inc.

# Introduction

Reliability Simulator performs reliability simulation to analyze the effects of reliability defects, such as Hot-Carrier Injection (HCI), Negative Bias Thermal Instability (NBTI)/ Positive Bias Thermal Instability (PBTI) or self-heating, on the circuit and predict the reliability of designs in terms of performance.

For more information about HCI, NBTI, and PBTI, see Spectre Reliability Analysis in *Spectre Classic Simulator, Spectre APS, Spectre X, and Spectre XPS User Guide*.

There are two analysis modes available:

- Spectre native
- RelXpert

**Important points to note:**

- Currently, reliability analysis is run with Sweeps and Corners and Monte Carlo run modes.
- Reliability analysis supports *stb*, sp, *pss*, *pnoise*, *hb*, and *hbnoise* analyses. These analyses are not supported when gradual aging is enabled.
- Stress simulations for reliability analysis require a transient analysis.

To run a reliability simulation, you need to perform the following tasks:

1. Specify Reliability options
2. Run the Reliability simulation
3. View the Reliability simulation results
4. View the Reliability aged netlist
5. Annotate simulation results to the schematic

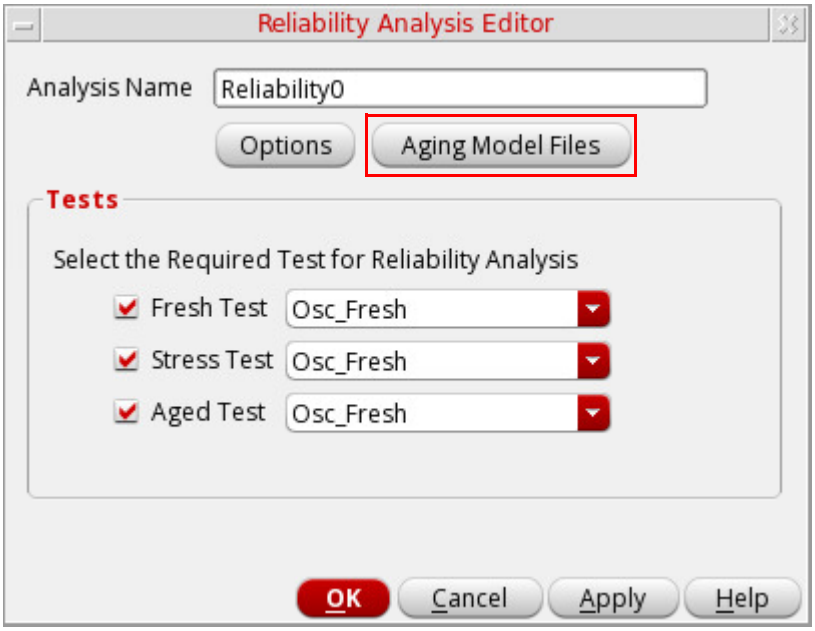
# Specifying Reliability Options

To specify Reliability options, follow the steps listed below:

1. Ensure that a reliability model is included and a transient analysis is set up.


© 2023 Cadence Design Systems, Inc.

2. In the Setup assistant pane, expand the *Reliability Analyses* node and click where it says *Click to add reliability analysis setup*.
   The *Reliability Analysis Editor* form opens.



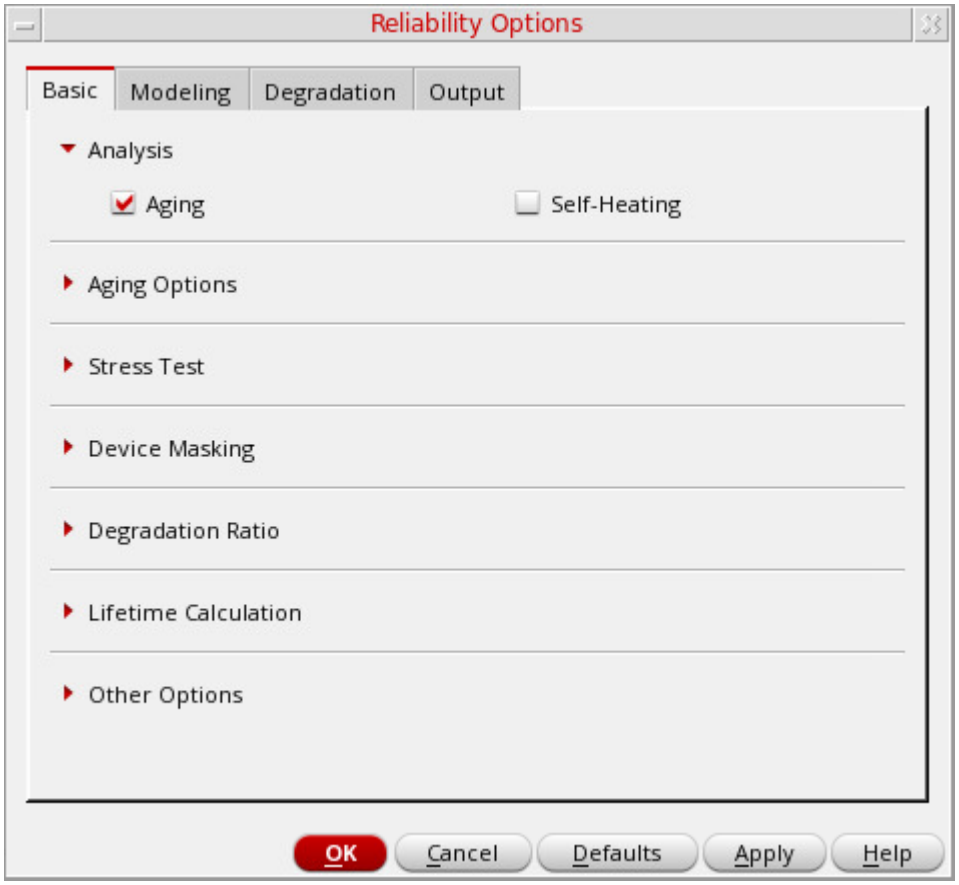

   Name of the selected test appears in the fields: *Fresh Test*, *Stress Test*, and *Aged Test*. The *Aging Model Files* command is displayed in the Reliability Analysis Editor form only if the displayAgingModelFiles environment variable is set to true. See the Specifying Aging Model Files section for more information.

   **Note:** If the *Stress Test* is enabled while setting up an EMIR analysis, the control statement `enable_emir_on_stress value=yes` gets added to the reliability control block in the netlist file.

3. Click *Options* to open the Reliability Options form to configure options for the reliability analysis. For more information about these options, see the Reliability Options Form section in the Virtuoso ADE Assembler User Guide.


© 2023 Cadence Design Systems, Inc.

The *Reliability Options* form appears.




4. Specify the required options and click *OK*.

For Spectre Simulator, setting up transient analysis is required before running stress simulation. However, aging simulations can be run with setting up either AC or DC analysis.

For running the reliability simulation with stress, you must specify the transient analysis, and then you can add AC or DC analyses if required.


© 2023 Cadence Design Systems, Inc.

After running the simulation, reliability options are netlisted as shown below:

```
rel reliability  {
    enable_ade_process value=yes
    age time=[10.0000y]
    tmi_aging_mode type=she
    tran_stress tran stop=1n write="spectre.ic" writefinal="spectre.fc"
    method=trap annotate=status maxiters=5
    finalTimeOP_stress info what=oppoint where=rawfile
    dc_stress dc param=temp start=1 stop=10 write="spectre.dc" \
    oppoint=rawfile maxiters=150 maxsteps=10000 annotate=status
    ac_stress ac start=1 stop=100 annotate=status
    tran_aged tran stop=1n write="spectre.ic" writefinal="spectre.fc" \
    method=trap annotate=status maxiters=5
    finalTimeOP_aged info what=oppoint where=rawfile
    dc_aged dc param=temp start=1 stop=10 write="spectre.dc" \
    oppoint=rawfile maxiters=150 maxsteps=10000 annotate=status
    ac_aged ac start=1 stop=100 annotate=status
    export E_T=oceanEval("1")
}
```

## Specifying Aging Model Files

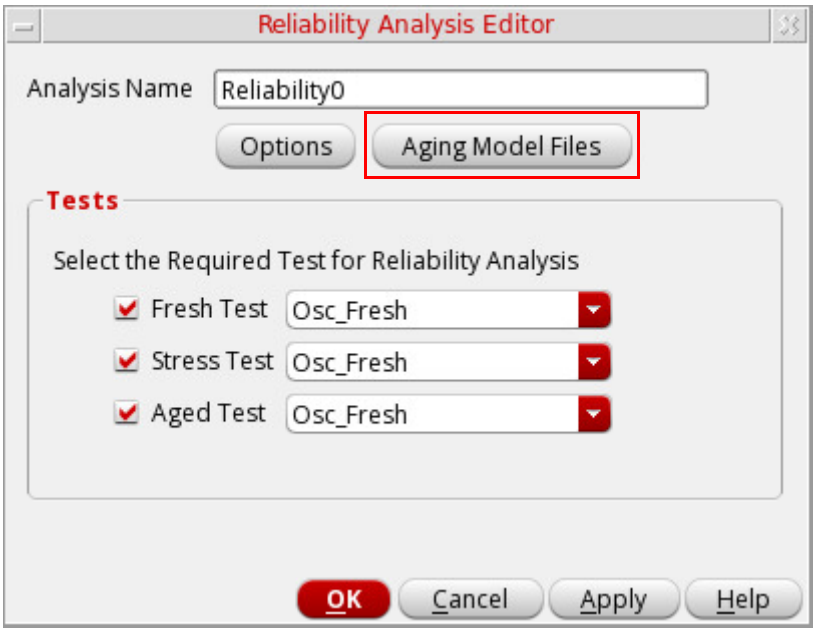
**Note:** Before specifying the aging model files, ensure that the following environment is set to true.

```
envSetVal("maestro.gui" "displayAgingModelFiles" 'boolean t)
```

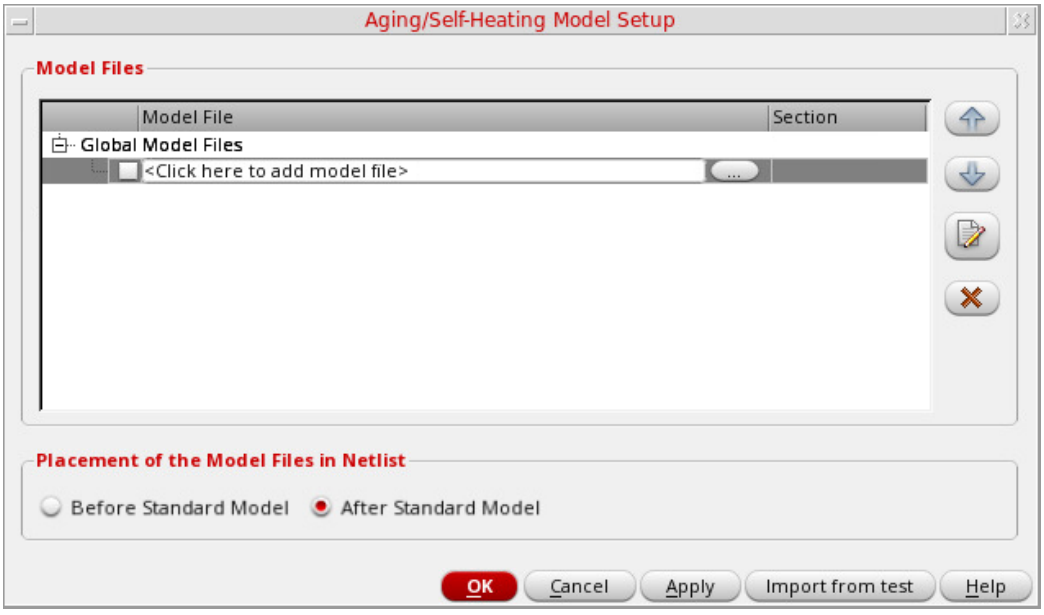
Perform the following steps to specify the model files to be used for aging or self-heating analysis for the stress or aged tests:


© 2023 Cadence Design Systems, Inc.

1. In the *Reliability Analysis Editor* form, click *Aging Model Files*, as shown in the following figure.




The the *Aging/Self-Heating Model 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 the ellipsis button (...) to open the *Choose Model File* form. Select a valid model file and click *Open*.

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


© 2023 Cadence Design Systems, Inc.

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.

4. In the *Placement of the Model Files in Netlist* section, select one of the following options to define whether you want to print the aging-specific model file before or after the standard model files in the netlist:

   - *Before Standard Model*: Prints the aging model file before the standard model file in the netlist
   - *After Standard Model*: Prints the aging model file after the standard model file in the netlist

   **Note:** You can also specify these options by setting the defaultAgingPlacementType environment variable.

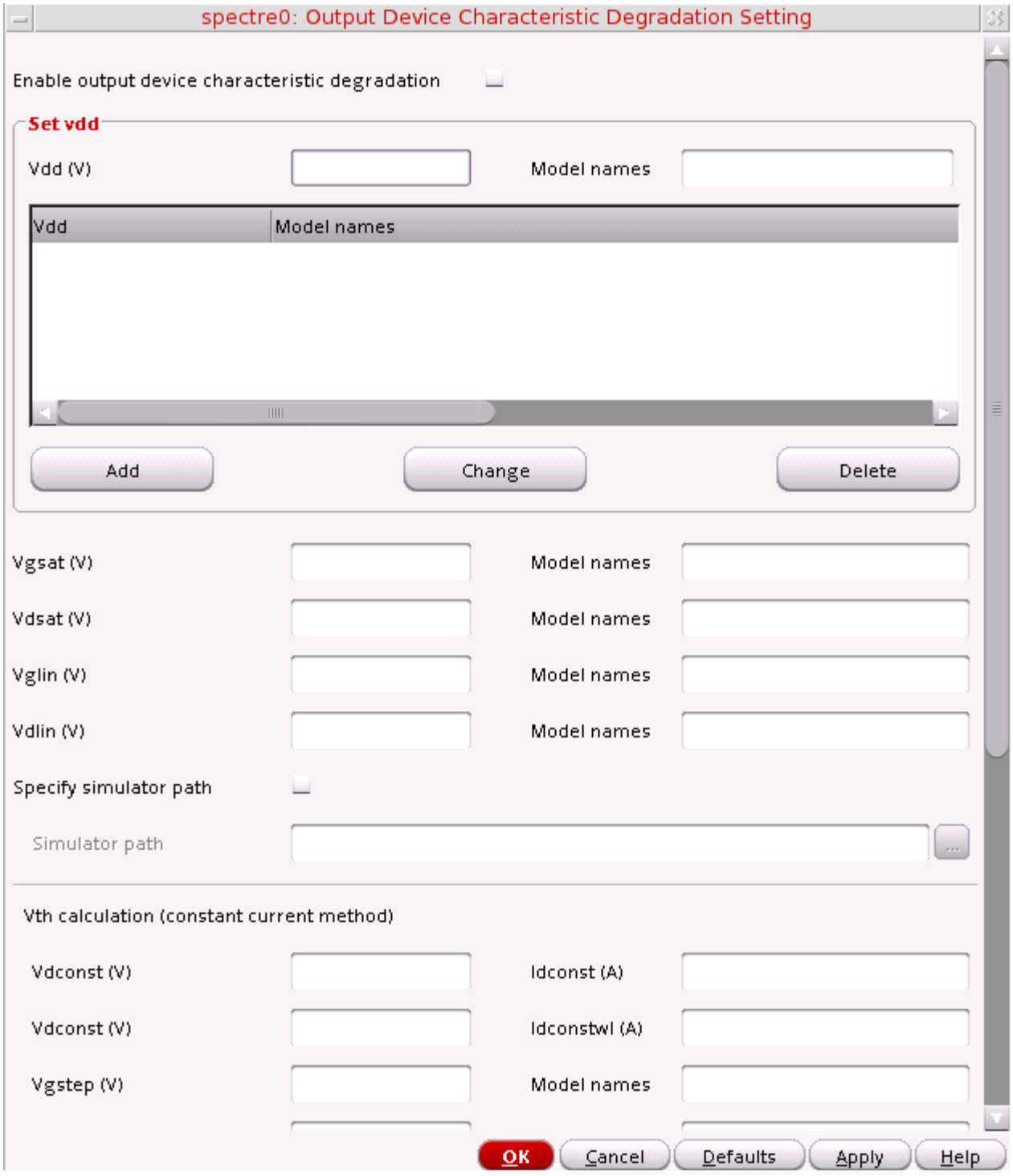
## Specifying Reliability Options

You can specify the reliability options in the *Reliability Options* form. For more information about these options, refer to Reliability Options Form.


© 2023 Cadence Design Systems, Inc.

## Output Device Characteristic Degradation Settings

The *Output Device Characteristic Degradation Settings* form contains options shown in the following figure.



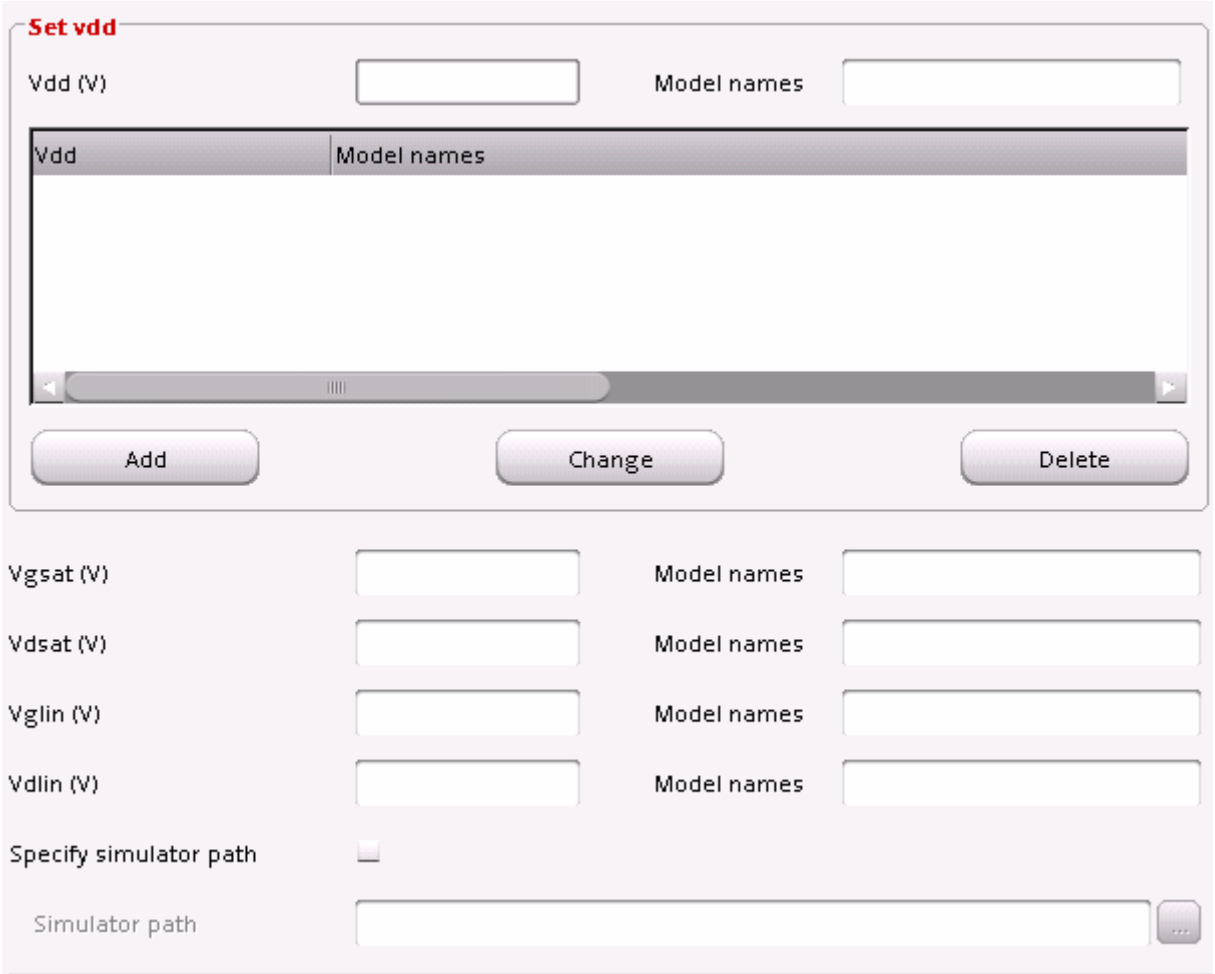


© 2023 Cadence Design Systems, Inc.

The options available on the *Output Device Characteristic Degradation Settings* form are described below.

- *Enable output device characteristic degradation*—Requests RelXpert Reliability Simulator to generate device degradation (gds, gm, Idlin, Idsat, Vth degradation) for the specified age time period.

### Set vdd




- *Vdd(V)*—Specifies the operation voltage for the circuit or target model.
- *Model names*—Specifies the model used in the .output_device_degrad file for the adjacent field. This field supports wildcards. For example, `n*` can be specified instead of `nmos1`.
- *Vdd*—Lists the added Vdd values in the table.

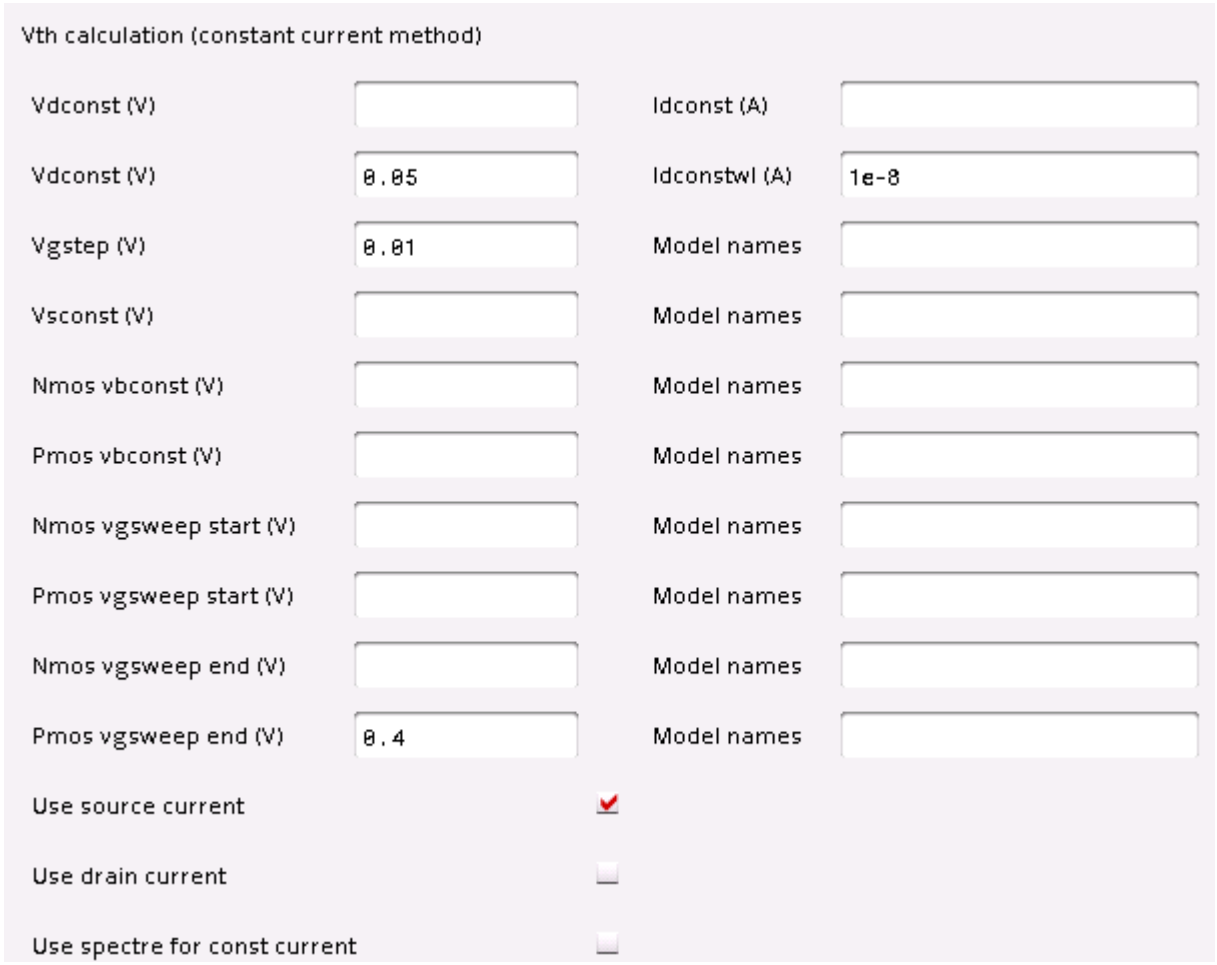

© 2023 Cadence Design Systems, Inc.

- *Model names*—Lists the model names specified for the added Vdd values.
- *Add*—Adds a new model specifying the operation voltage. Multiple model names may be added.
- *Change*—Updates the modifications after a particular model in the table is modified.
- *Delete*—Deletes the selected row.
- *Vgsat(V)*—Specifies Vgs value for Idsat measurement. The path to the corresponding model files must be specified in the Model names field.
- *Vdsat(V)*—Specifies Vds value for Idsat measurement. The path to the corresponding model files must be specified in the Model names field.
- *Vglin(V)*—Specifies Vgs value for Idlin/Vt/Gm measurement. The path to the corresponding model files must be specified in the Model names field.
- *Vdlin(V)*—Specifies Vds value for Idlin/Vt/Gm measurement. The path to the corresponding model files must be specified in the Model names field.
- *Specify simulator path*—If selected, enables or disables the specified simulator path.
- *Simulator path*—Specifies the full path for the simulator that is used during aging simulation.

**Note:** For more information about the above options, see Spectre Reliability Analysis in *Spectre Classic Simulator, Spectre APS, Spectre X, and Spectre XPS User Guide*.


© 2023 Cadence Design Systems, Inc.

## Vth calculation (constant current method)




- *Vdconst(V)*—Specifies the constant drain voltage for calculating the threshold voltage by using the Vt_const_current method.
- *Idconst(A)*—Specifies the constant drain current for calculating the threshold voltage by using the Vt_const_current method.
- *Idconstwl(A)*—Specifies the constant drain current related to the transistor's width(W) and length(L).
- *Vgstep(V)*—Specifies the gate voltage (Vg) sweep step size for calculating the device degradation(Vth, ids, idling, gds, and gm degradation). The path to the corresponding model files must be specified in the Model names field.
- *Vsconst(V)*—Specifies the constant source terminal voltage for calculating the threshold voltage degradation. The path to the corresponding model files must be specified in the Model names field.


© 2023 Cadence Design Systems, Inc.

- *Nmos vbconst(V)*—Specifies the constant substrate terminal voltage for NMOSFET for calculating the threshold voltage degradation. The path to the corresponding model files must be specified in the Model names field.
- *Pmos vbconst(V)*—Specifies the constant substrate terminal voltage for PMOSFET for calculating the threshold voltage degradation. The path to the corresponding model files must be specified in the Model names field.
- *Nmos vgsweep start (V)*—Specifies the start point of the sweep voltage of the NMOSFET gate terminal for calculating the threshold voltage degradation. The path to the corresponding model files must be specified in the Model names field.
- *Pmos vgsweep start (V)*—Specifies the start point of the sweep voltage of the PMOSFET gate terminal for calculating the threshold voltage degradation. The path to the corresponding model files must be specified in the Model names field.
- *Nmos vgsweep end (V)*—Specifies the stop point of the sweep voltage of the NMOSFET gate terminal for calculating the threshold voltage degradation. The path to the corresponding model files must be specified in the Model names field.
- *Pmos vgsweep end (V)*—Specifies the stop point of the sweep voltage of the PMOSFET gate terminal for calculating the threshold voltage degradation. The path to the corresponding model files must be specified in the Model names field.
- *Use source current*—If selected, enables or disables the drain current to calculate the degraded threshold voltage with vt_const_current method.
- *Use drain current*—If selected, enables or disables the source current to calculate the degraded threshold voltage with vt_const_current method.
- *Use spectre for const current*—If selected, enables or disables the const current method to calculate the threshold degradation value.

**Note:** For more information about the above options, see Spectre Reliability Analysis in *Spectre Classic Simulator, Spectre APS, Spectre X, and Spectre XPS User Guide*.

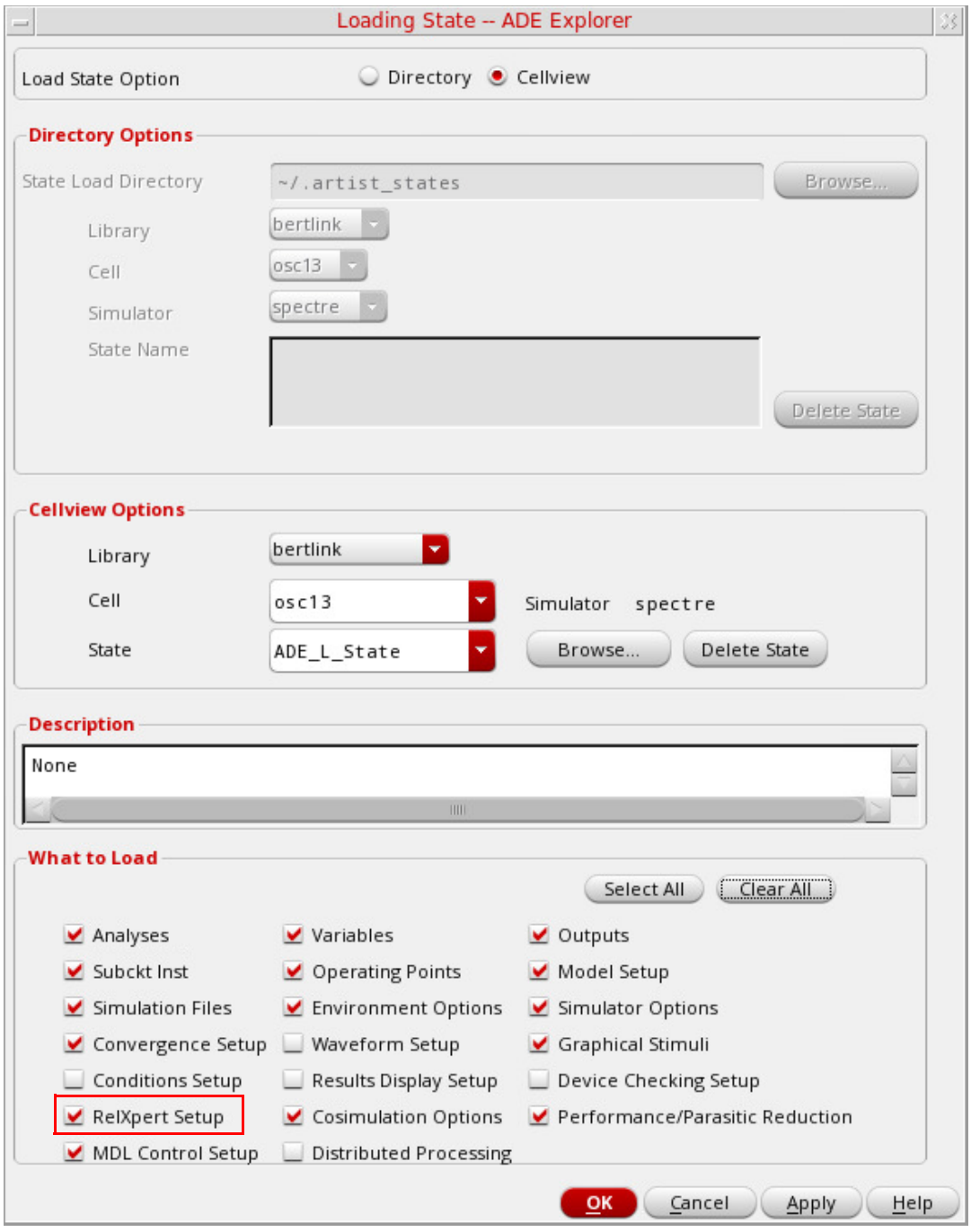
# Migrating Reliability Setup from ADE L

If you have defined a reliability setup in an ADE L state, you can migrate it to Explorer by following these steps:

1. From the *Menu* bar, choose *Session – Load ADE L State* to load the previously saved ADE L state in the current session.


© 2023 Cadence Design Systems, Inc.

The *Loading Sate – ADE Explorer* form appears.



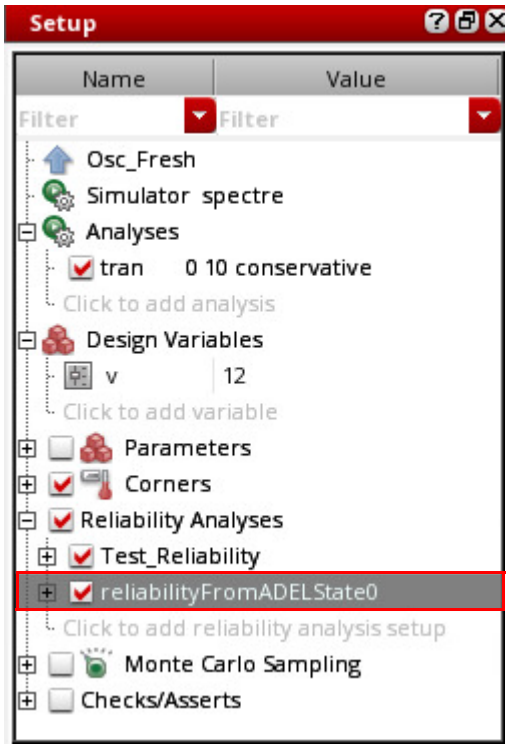

2. From the *Load State Option*, select *Cellview*.
3. In the *Cellview Options* section, select the ADE L state from which you want to migrate the reliability setup by defining library, cell, and state.
4. In the *What to Load* section, select the *RelXpert Setup* check box.


© 2023 Cadence Design Systems, Inc.

5. Click *OK*.

   The reliability setup from the ADE L state is migrated to the current Explorer session.

6. To view the migrated setup, expand the *Reliability Analyses* node in the *Setup* assistant, as shown in the following figure.




   **Note:** The default name of the migrated reliability setup is `reliabilityFromADELState#`.

You can also load the reliability settings from one reliability setup to another reliability setup. For more information, see the Loading Reliability Settings section.

# Running the Reliability Simulation

Perform the following steps to run a reliability simulation:

1. Ensure that the following is done:
   - ❑ Model libraries are included in the reliability model
   - ❑ A transient analysis is set up and enabled before running stress simulation
   - ❑ The *Enable Analysis* check box is selected in the *Reliability* form


© 2023 Cadence Design Systems, Inc.

2. Choose *Simulation – Run* or *Netlist and Run* or click the run button, .

***Important Points to Note:***

- Reliability Simulation always runs in batch mode. Therefore, the ADE Explorer mode too is automatically set to batch when you enable Reliability, and the following message is displayed in CIW:

  ```
  *WARNING* The Spectre run mode needs to be 'batch' when running
  RelXpert. Automatically setting the run mode to 'batch'.
  ```

- ADE Explorer does not support running multiple reliability analyses simultaneously. Ensure that only one reliability analysis is enabled at a time or use ADE Assembler to run multiple reliability analyses simultaneously.

# Viewing the Reliability Simulation Results

Right-click in a row on the *Results* tab and choose *Reliability Data* to access the Reliability simulation results. You can use this menu to view the following results:

- Device Lifetime and Degradation Results
- Model Parameter Changes Results

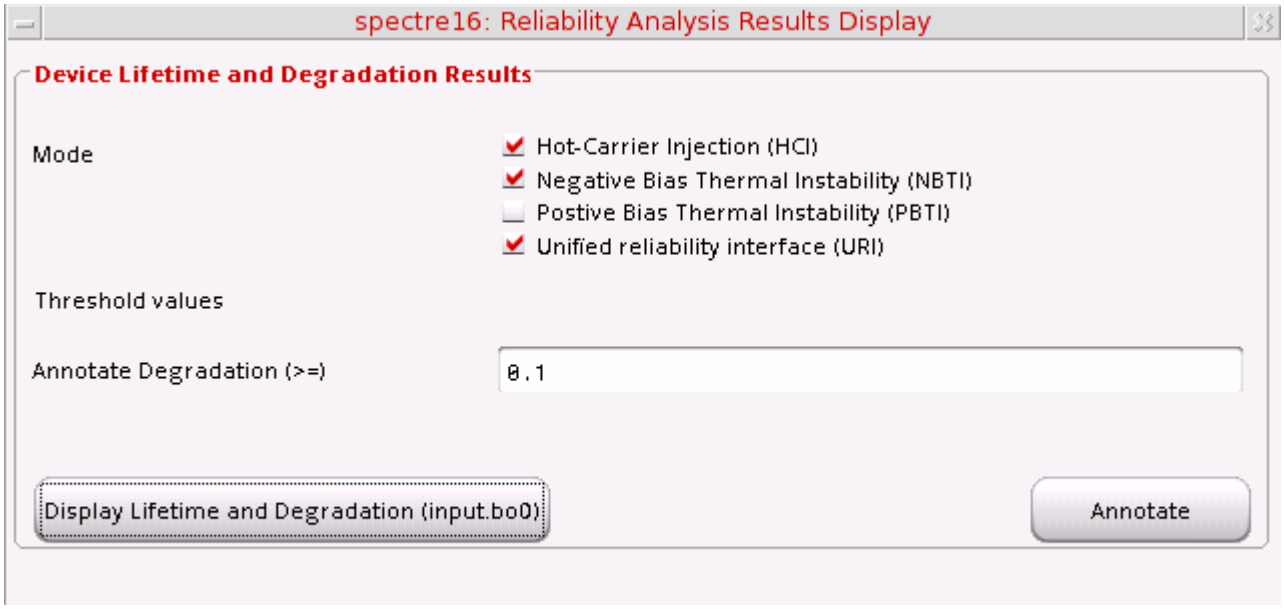
## Device Lifetime and Degradation Results

Perform the following steps to view the Device Lifetime and Degradation results:

➥ Right-click in a row on the *Results* tab and choose *Reliability Data – Device Lifetime and Degradation*.


© 2023 Cadence Design Systems, Inc.

The *Device Lifetime and Degradation Results* form appears.



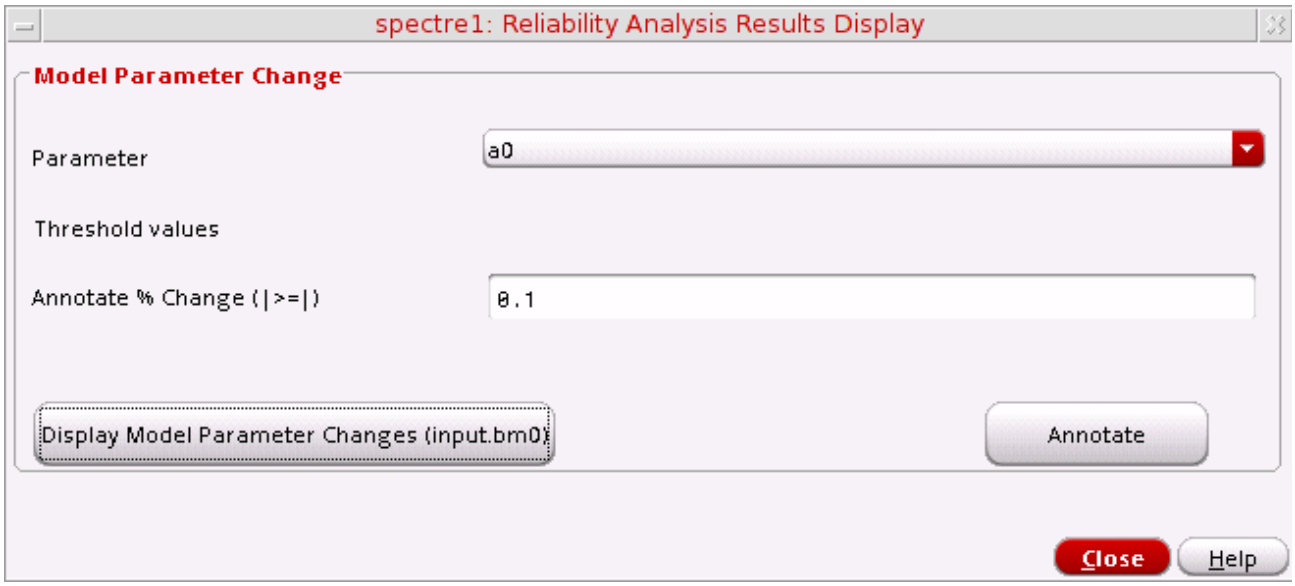

## Model Parameter Changes Results

Perform the following steps to view the Model Parameter Changes Results:

➡ Right-click in a row on the *Results* tab and choose *Reliability Data – Model Parameter Changes.*


© 2023 Cadence Design Systems, Inc.

The *Model Parameter Changes Results* form appears displaying the fresh and aged SPICE model parameter information.




**Note:** This result is available only if you select the *Report model parameter changes* check box on the *Advanced* tab.

# Viewing the Reliability Aged Netlist

The aged netlist can be used to view the degradation data generated by RelXpert Reliability Simulator.

**Note:** This option is available only if you select *RelXpert* as the *Simulator Mode* on the Reliability form.

To view the aged netlist, do the following:

➡ Right-click in a row on the *Results* tab and choose *Reliability Data – Aged netlist*.

The Reliability Simulator generated degradation netlist is displayed.

# Annotating Simulation Results to the Schematic

Perform the following steps to annotate the Device Lifetime and Degradation Results to the schematic:


© 2023 Cadence Design Systems, Inc.

1. Right-click in a row on the *Results* tab and choose *Reliability Data – Device Lifetime and Degradation Results*.

   The *Device Lifetime and Degradation Results* form appears. The threshold rules corresponding to the selected modes are displayed in the form.

2. Select the threshold rules based on which you want to annotate the results.

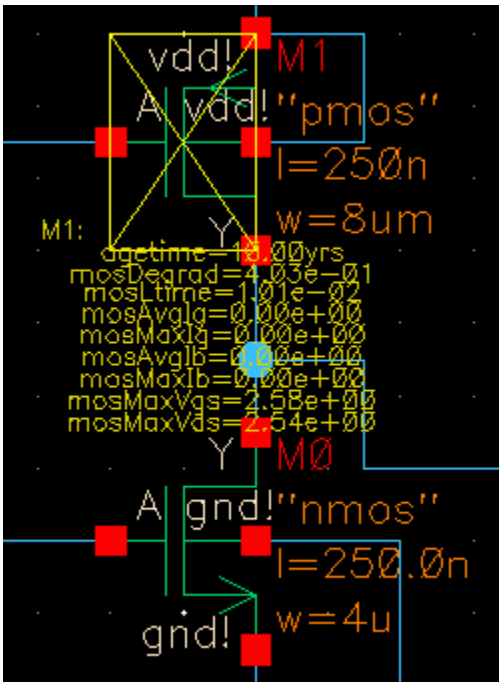
   The fields containing the default threshold values for the selected rules become available.

   **Note:** Results are annotated based on the specified threshold values. For example, if the *Lifetime (year) (<=)* field specifies a threshold value of 10, results are annotated only to those instances whose lifetime is less than or equal to 10 years.

3. Modify the threshold values for the selected rules, if required.

4. Click *Annotate*.

   The instances to which the results are annotated are highlighted in yellow on the schematic.




**Note:** In a hierarchical design, a block is highlighted if results are annotated to an instance in that block. When you descend into the block, the instances on which the results are annotated are also highlighted.

Similarly, you can annotate other types of reliability results.


© 2023 Cadence Design Systems, Inc.

# Saving Reliability Options

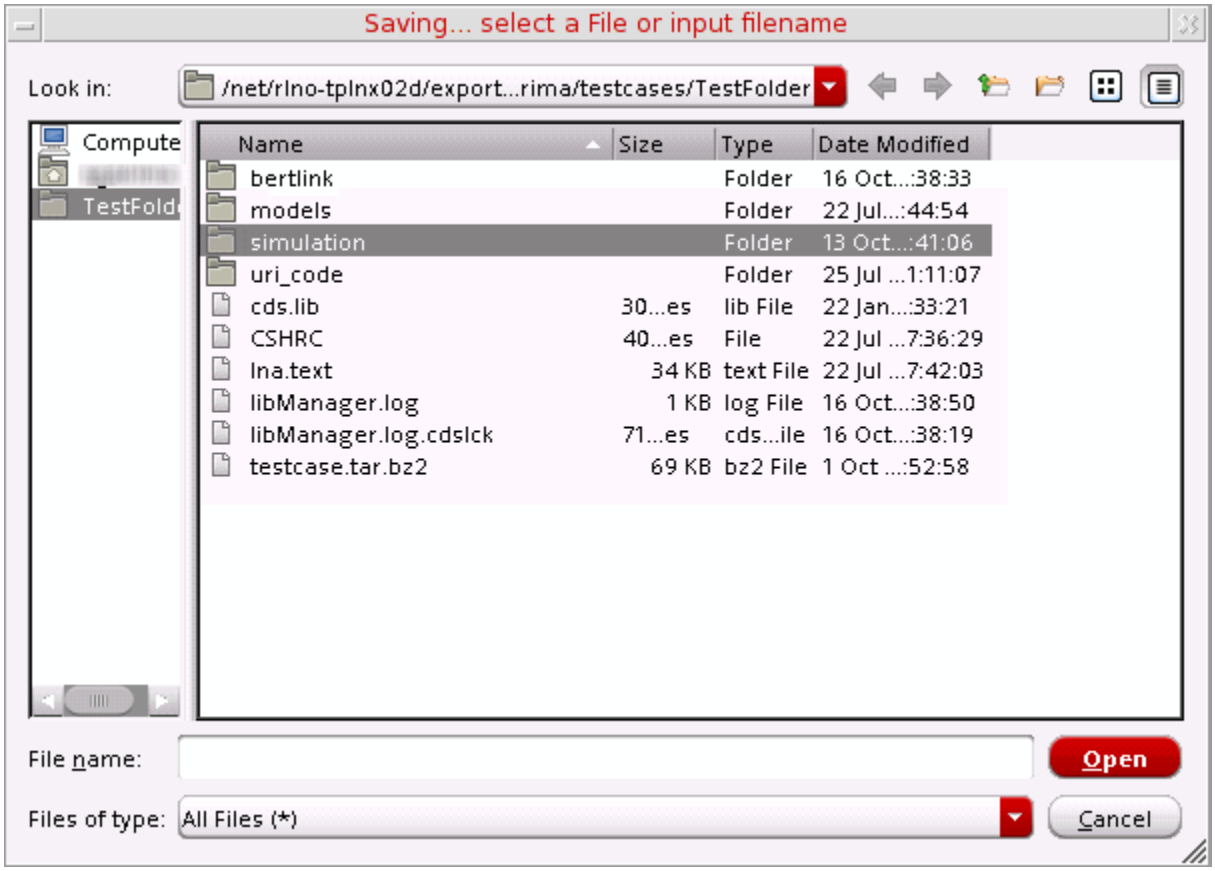
You can save the values of reliability options specified in the Reliability form to a text file and later load these values back into Reliability form.

Perform the following steps to save reliability options specified in the *Reliability* form to a text file.

1. Choose *Simulation – Reliability – Save Options*.

   Alternatively, right-click the test name in the Setup assistant and choose *Reliability – Save Options*.

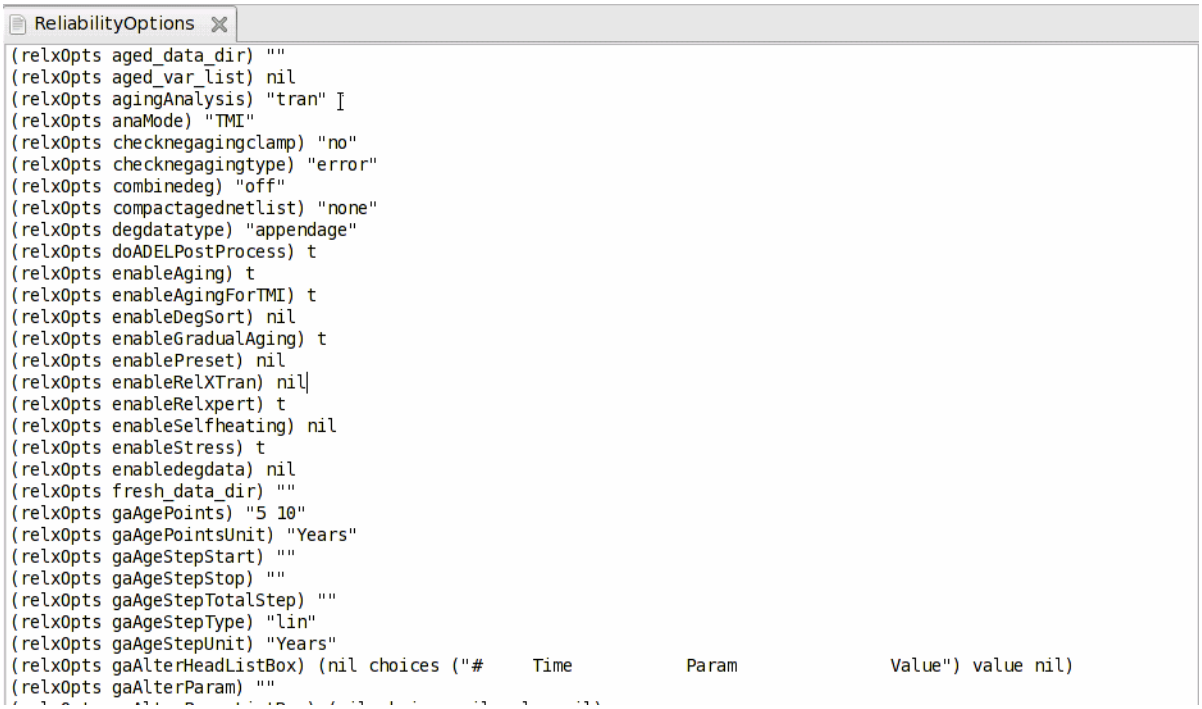
   The *Saving* form appears.




2. Specify the directory and type a name for the file.
3. Click *Save*.


© 2023 Cadence Design Systems, Inc.

A text file containing the reliability options specified in the *Reliability f*orm is created in the specified directory. You can edit this file in your default text editor.




# Loading Reliability Options

Perform the following steps to load the values of reliability options saved in a text file in the *Reliability* form.

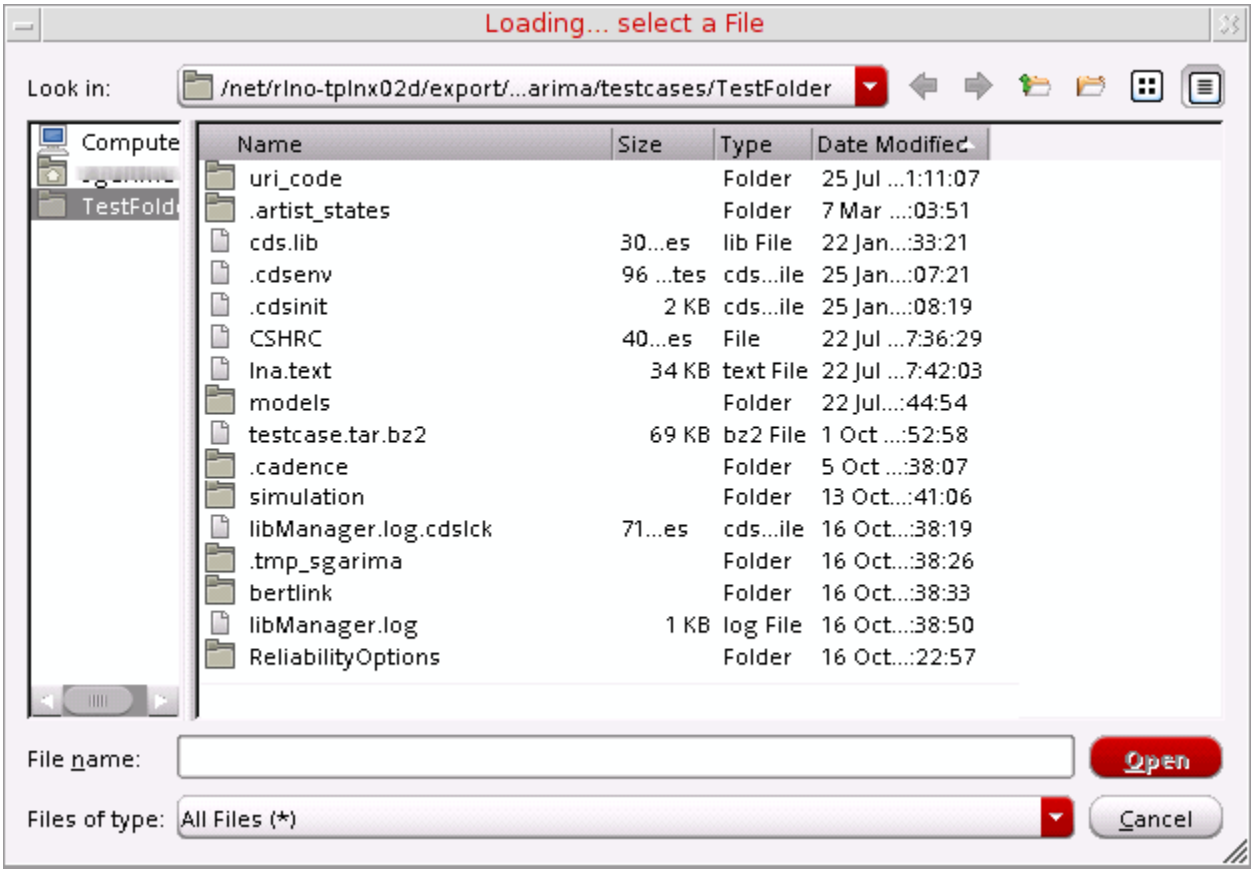
**Note:** For more information on how to load an existing Reliability setup, see Loading State.

1. Choose *Simulation – Reliability – Load Options*.

   Alternatively, right-click the test name in the Setup assistant and choose *Reliability—Load Options*.


© 2023 Cadence Design Systems, Inc.

The *Loading* form appears.




2. Browse and select the file containing the values of reliability options.
3. Click *Open*.

   The values of the reliability options specified in the text file are loaded in the *Reliability* form. To open the form and verify the values, choose *Reliability – Setup* to open the *Reliability* form.


© 2023 Cadence Design Systems, Inc.

© 2023 Cadence Design Systems, Inc.

# 13

# Performing Circuit and Device Checks

ADE Explorer provides an interactive user interface where you can define circuit or device checks that are performed using the simulation results. The violations of these checks are highlighted on the `Checks/Asserts` view on the Results tab. You can view and analyze the violations by using the various features of this view, such as, filtering violations using the built-in or custom filters, finding the critical instances or nets on the schematic, or highlighting the violation results on the waveforms. Next, waive or resolve the violations as per your requirements, and rerun simulations to verify the results.

This chapter describe how to use the tools and features in ADE Explorer to perform circuit and device checks, and to resolve violations:

- An Introduction to Checks and Asserts
- Adding Checks and Asserts
- Filtering Violations
- Running Simulations and Viewing Violations


© 2023 Cadence Design Systems, Inc.

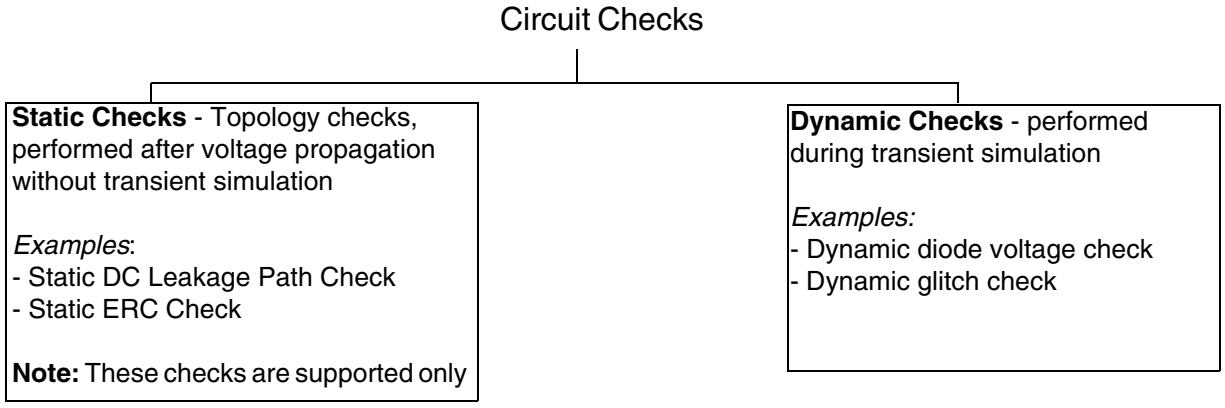
# An Introduction to Checks and Asserts

This section provides an introduction to the types of checks supported by ADE Explorer. It also introduces the *Checks/Asserts* tree in the Setup assistant.

## Checks

Checks (or circuit checks) enable you to analyze typical design problems, such as high impedance nodes, leakage paths between power supplies, setup and hold timing errors, power issues, connectivity problems, or extreme rise and fall times.

Circuit checks are of two types shown below.




For more details on the circuit checks that can be performed by the Spectre simulator, refer to Circuit Checks in *Spectre Classic Simulator, Spectre APS, Spectre X, and Spectre XPS User Guide*.

## Asserts

Asserts (or device checks) are used to perform checks on design parameters, node voltages, element currents, model parameters, operating point parameters, and expressions. In Spectre, the `assert` and `checklimit` statements are used to define and run device checks.

For details and examples on the device checks that can be performed by the Spectre simulator, refer to Device Checks in *Spectre Classic Simulator, Spectre APS, Spectre X, and Spectre XPS User Guide*.


© 2023 Cadence Design Systems, Inc.

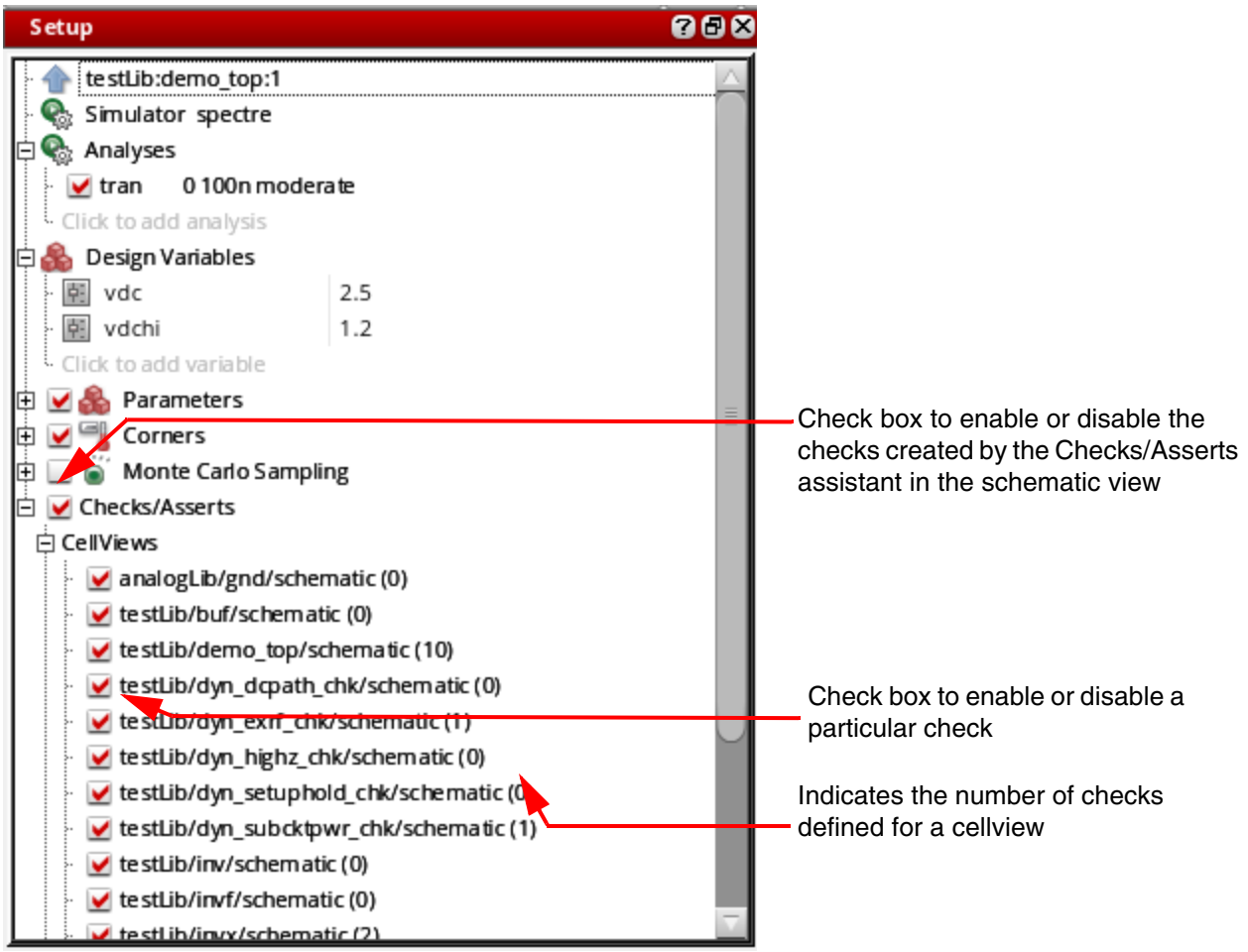
## Dependency on Simulators

In ADE Explorer, you can perform all the circuit and device checks that are supported by the MMSIM 14.1 ISR11 or later releases. The checks should be defined using the required syntax and keywords, and can include wildcards, subcircuits, and instance scoping. Spectre writes the results of these checks into the SQLite database in XML format. ADE Explorer reads that data and displays it in tabular format on the *Results* tab.

**Note:** The static checks are supported only in APS and XPS.

## Viewing Checks and Asserts in the Setup Assistant

The *Checks/Asserts* tree in the Setup assistant shows the cellviews for which checks and asserts have been defined. By default, this check box is disabled.





© 2023 Cadence Design Systems, Inc.

You can enable or disable checks for selected cellviews by selecting or clearing the check boxes next to them.

You can define checks for cellviews in the Checks/Asserts assistant. For more details, see Adding Checks and Asserts.

# Adding Checks and Asserts

Checks can be added in the following two ways:

- By using the Checks/Asserts assistant in the Virtuoso Schematic Editor window
- Copying Checks or Asserts
- By importing asserts from included text files

*Important*

The *Checks/Asserts* check box in the Setup assistant can enable or disable only the checks that are created using the Checks/Asserts assistant in the Schematic Editor, not the checks imported from the text files included in the Explorer setup.

**Note:** Any custom or default system constraint created in the Constraint Manager assistant can be propagated automatically to the Checks/Asserts assistant. For information about how to enable the automatic propagation, see Propagating Constraints to the Checks/Asserts Assistant in *Virtuoso Unified Custom Constraints Configuration Guide*.

## Adding Checks using the Checks/Asserts Assistant

To add checks, perform the following steps:

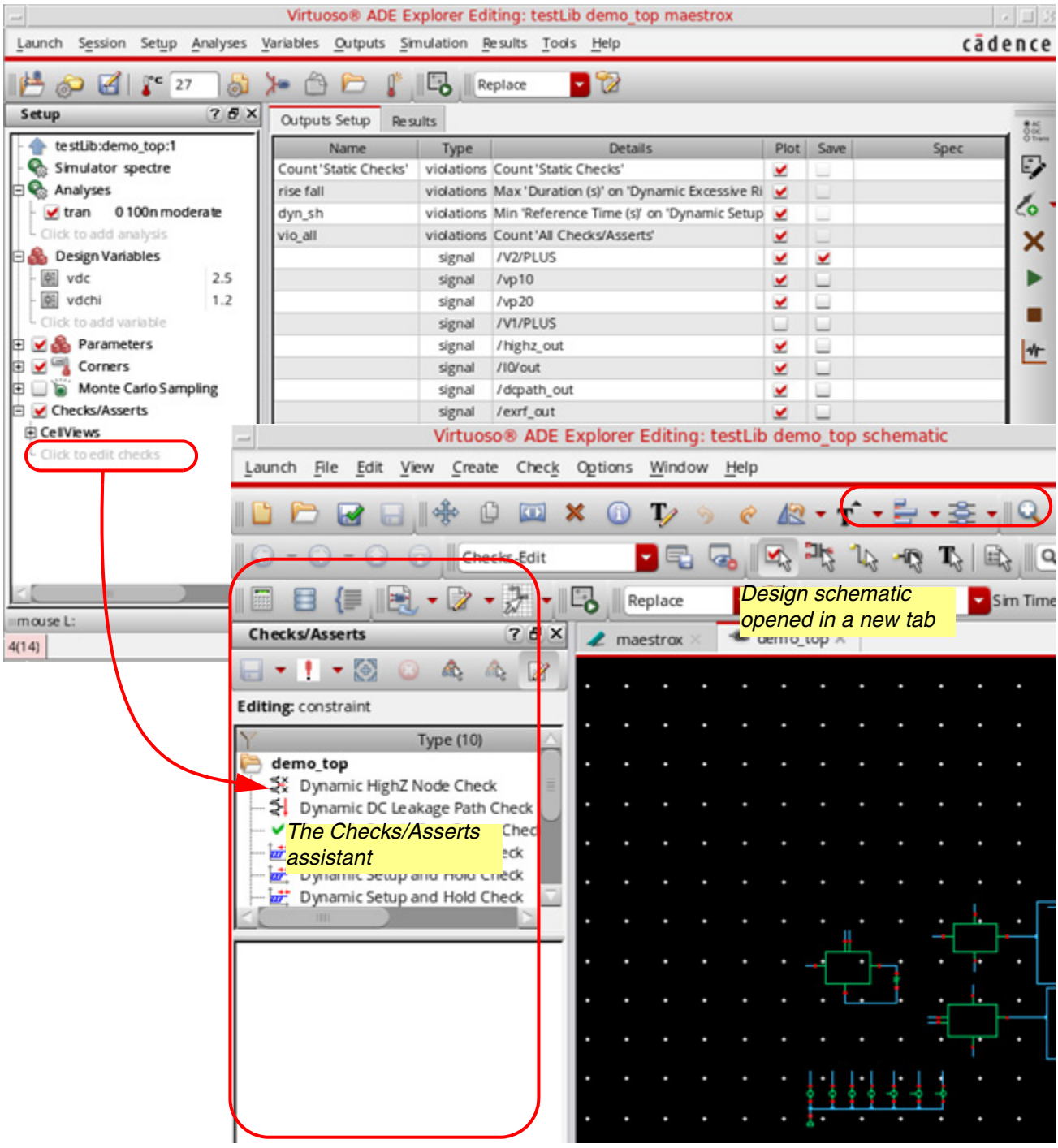
1. In the Setup assistant, expand the *Checks/Asserts* tree.

   The Cellviews tree appears including all the available cellviews.

2. Click *Click to edit checks*.


© 2023 Cadence Design Systems, Inc.

The schematic view of that cell is opened in a new tab.



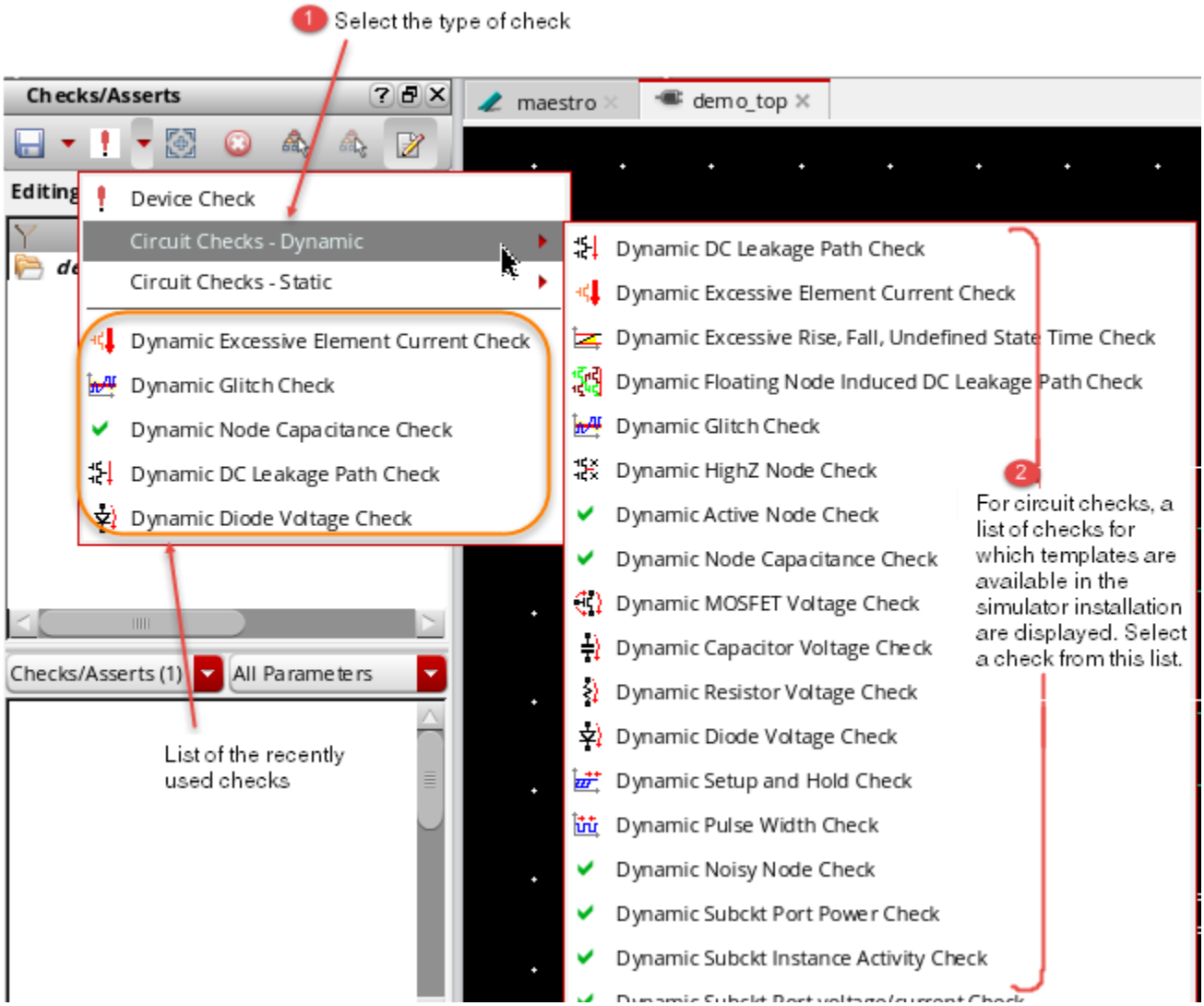

By default, the `Checks-Edit` workspace is opened. This workspace shows the *Checks/Asserts* assistant on the left.


© 2023 Cadence Design Systems, Inc.

To add new checks, perform the following steps:

1. In the Checks/Asserts assistant, click the checks drop-down list, and choose the type of check you need to create. For example, *Circuit Checks - Dynamic*.

   The list of dynamic checks for which templates are available in Spectre are shown in the submenu.




   The check is added to the top section of the *Checks/Asserts* assistant. Its corresponding parameters are displayed in the bottom section, which is also called as the Editor pane of this assistant.

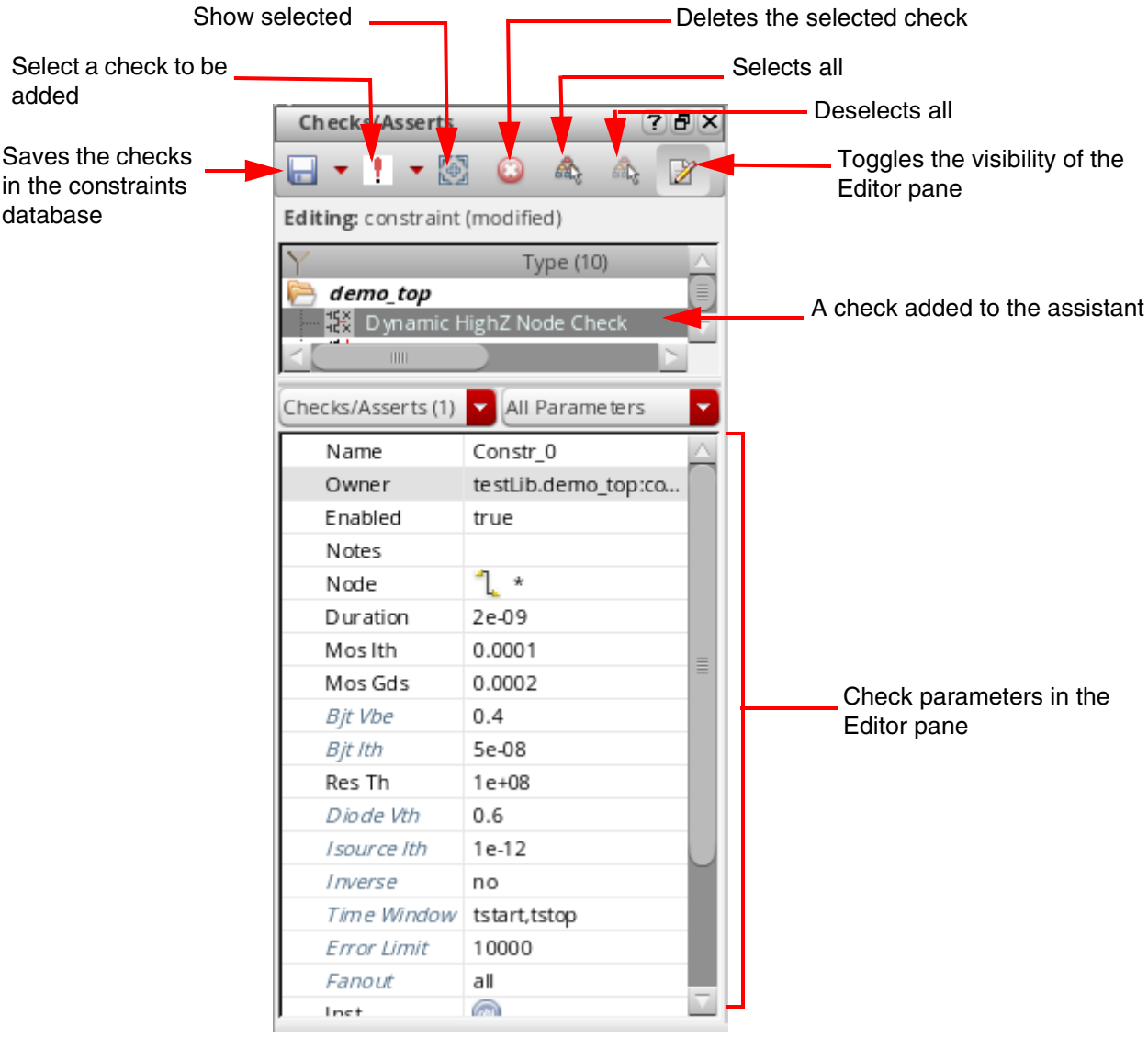
   There is no list of checks for *Device Check*. For this type of checks, you can directly provide parameters in the Editor pane.


© 2023 Cadence Design Systems, Inc.

2. Move the pointer over a check name in the top area of the assistant to view its definition and parameters in a tooltip.

*Tip*

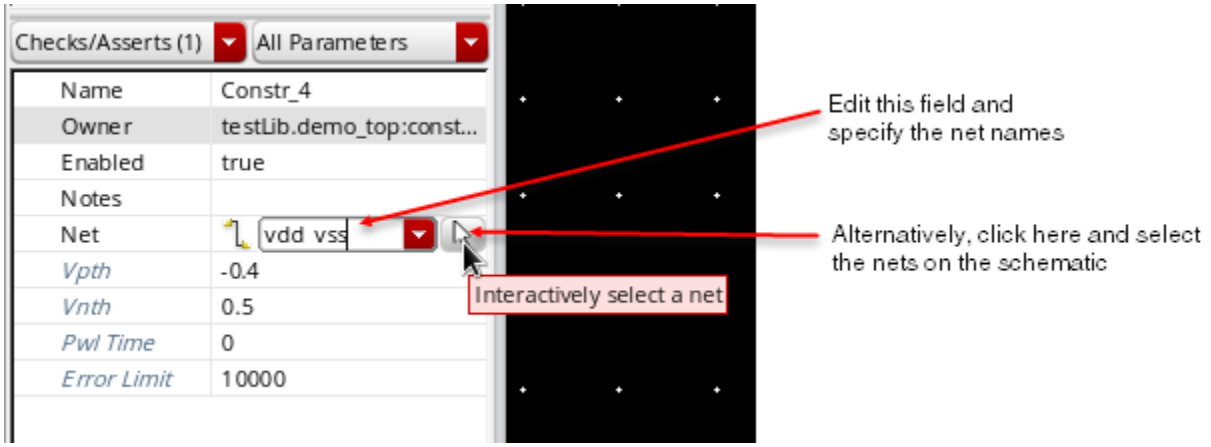
If the Editor pane is not visible, click *Checks/Asserts Editor* ( ) on the assistant toolbar.




3. Edit the check parameters as required. You can define the scope for a check by setting the *Node*, *Device*, and *Inst* fields.


© 2023 Cadence Design Systems, Inc.

Either type the names of nodes, devices, and instances, or directly select these from the schematic.




To apply a check for all the nodes or instances, enter `*` in the respective field.

If a field value for a parameter is unchanged, it will use the default values for that parameter as defined in the simulator file. The default values can be found in two ways:

- By using the `spectre -help <checkName>` command, as shown below.

  ```
  spectre -help assert
  spectre -help dyn_highz
  ```

- By moving the pointer over any check parameter in the Editor pane. The definition and default value for a parameter is displayed in the tooltip.

4. Repeat steps 1 to 3 to add more checks.
5. Click *Save* in the toolbar to save the checks.

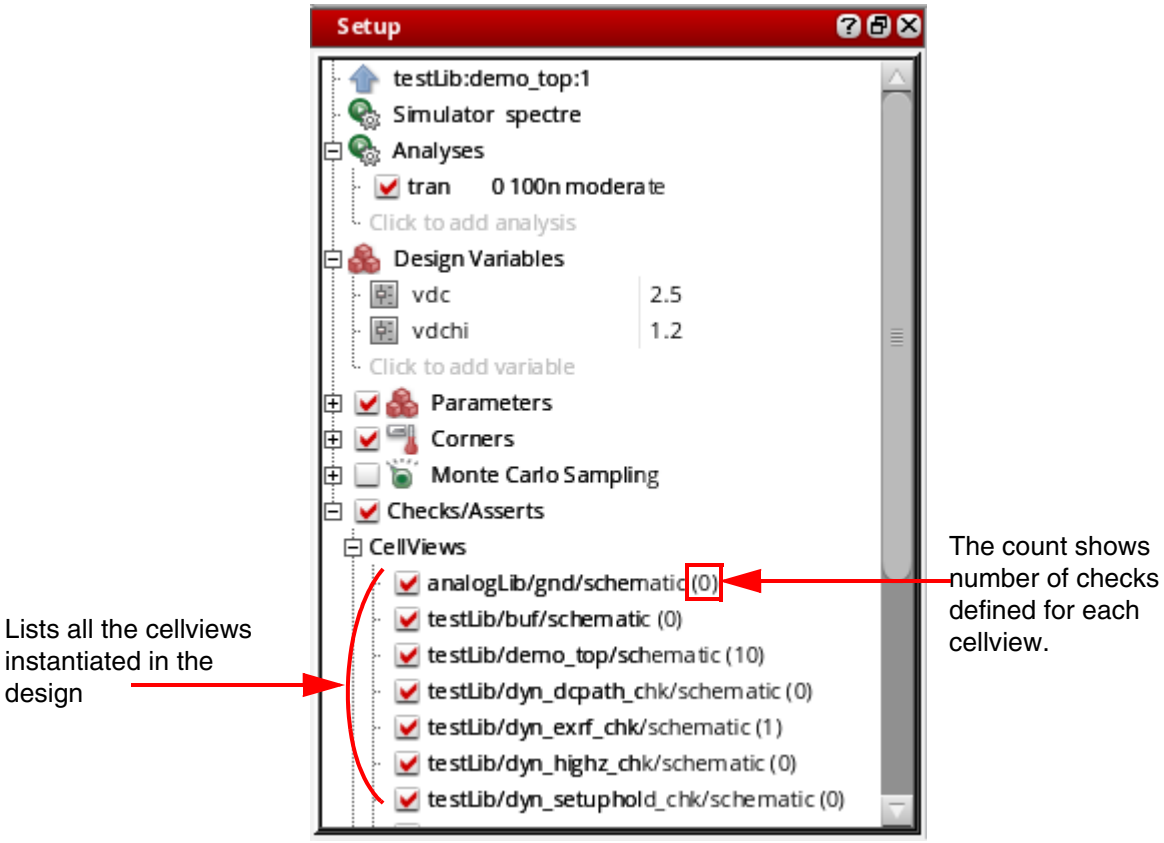
*Important*

It is required to save the checks to ensure they are netlisted correctly. If you run simulation without saving changes in the constraints view, ADE Explorer displays a message and prompts you to save changes.


© 2023 Cadence Design Systems, Inc.

*Tip*

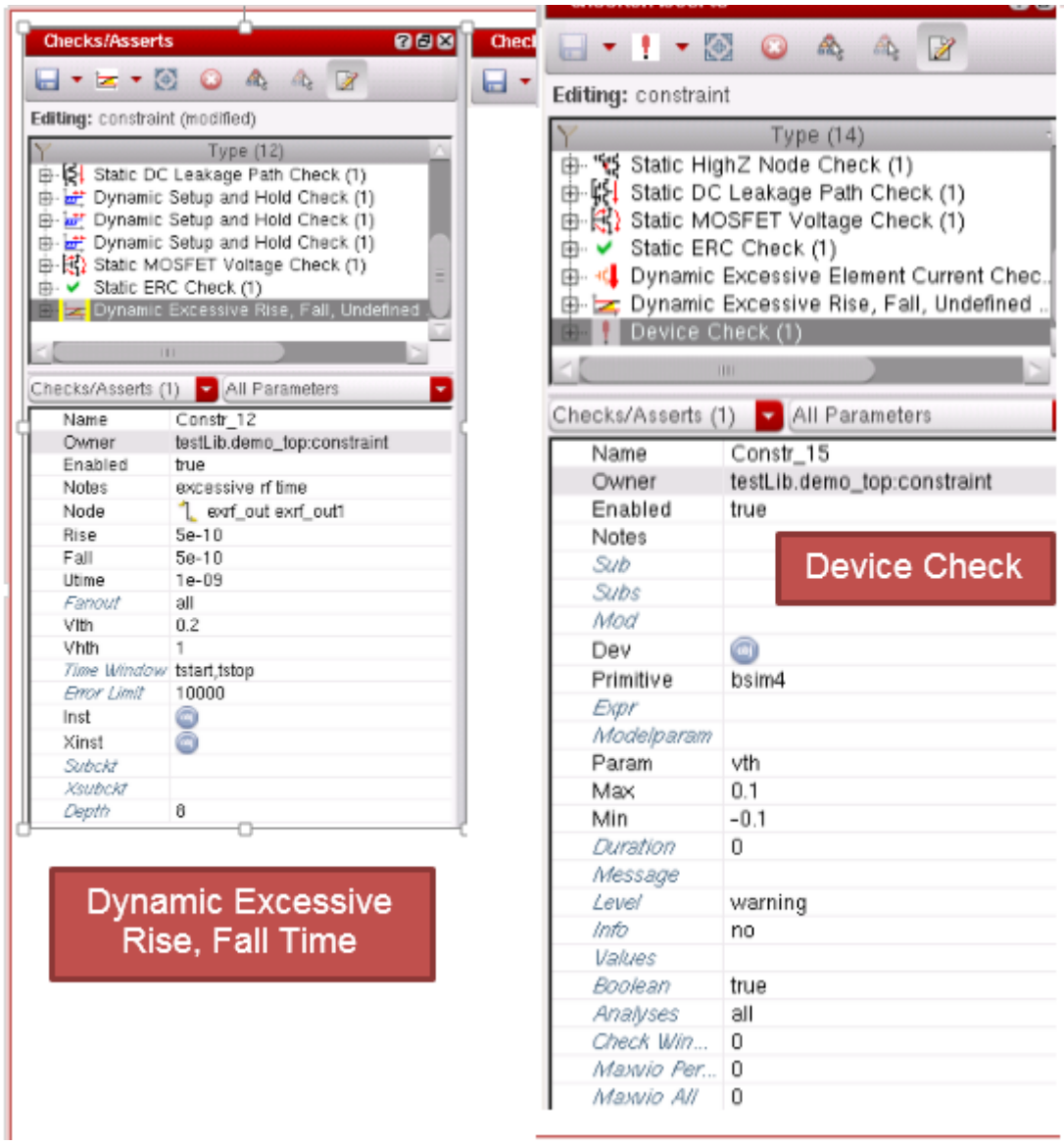
The *Checks/Asserts — Cellviews* tree lists all the cellviews contained in the design. The numerical value next to each cellview shows the count of checks defined for each. To edit the checks for a cellview, right-click it and choose *Edit Checks*. The cellview is opened in the schematic view where you can add or edit checks in the Checks/Asserts assistant.





© 2023 Cadence Design Systems, Inc.

An example displaying a circuit check and a device check is shown below.




*Important*

The checks and asserts defined for a cell using the *Checks/Asserts* assistant are stored in a special partition of the constraint view for the cell. These checks will be netlisted using the subcircuit parameter to keep the scope of the check to the cellview in which it is created. These checks are included in the simulation whenever that cell is instantiated in a design.

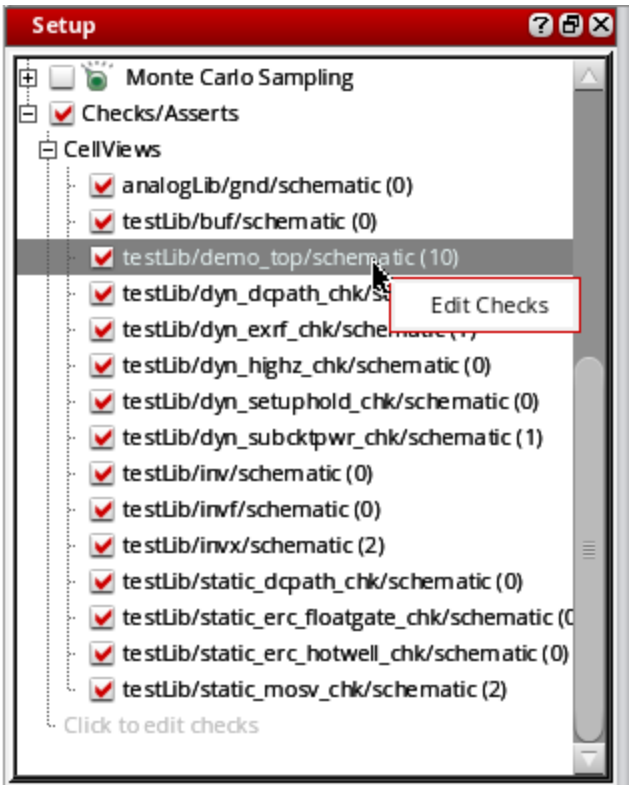

© 2023 Cadence Design Systems, Inc.

## Copying Checks or Asserts

If the check or assert you need to create for a cellview already exists for another cellview instantiated in the design, you can copy that and customize it, if required.

To copy a check or assert, perform the following steps:

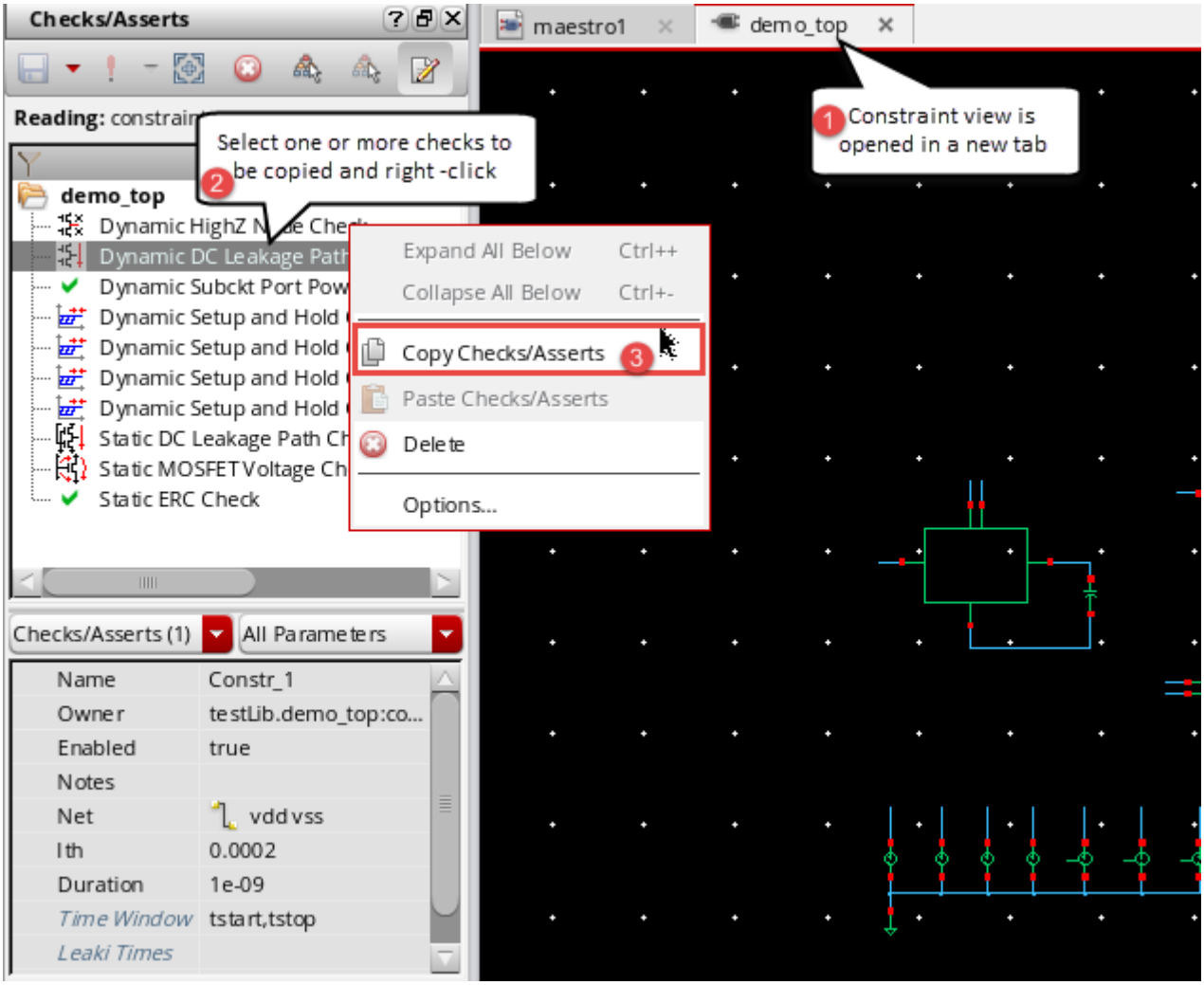
1. Right-click the source cellview from which you need to copy the check or assert, and choose *Edit Checks*.




   The constraint view of the cellview is opened in a new tab and the checks/Asserts assistant is displayed.


© 2023 Cadence Design Systems, Inc.

2. Right-click one or more checks and asserts to be copied and choose *Copy Checks/Asserts* from the context-sensitive menu.

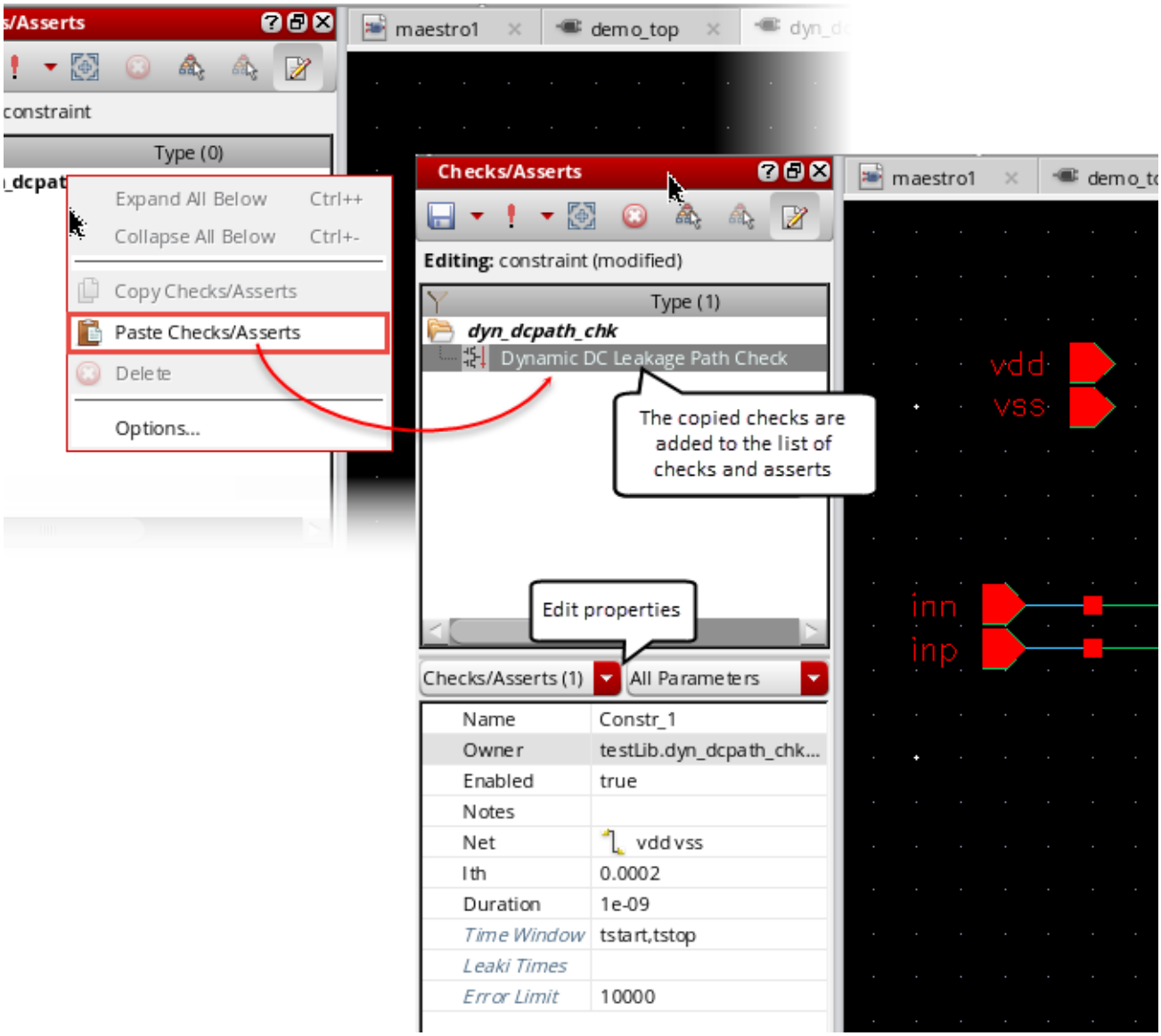



The copied check or assert is saved in the memory.

3. Reopen the `maestro` cellview tab.
4. Right-click the cellview to which you need to copy or reuse the copied check or assert, and choose *Edit Checks*.
5. The constraint view of the target cellview is also opened in a new tab and the Checks/Asserts assistant is displayed
6. Ensure that the constraint view is editable. If not, click *Save Checks/Asserts – Make Editable*.
7. Right-click anywhere in the checks/asserts pane and choose *Paste Checks/Asserts*.


© 2023 Cadence Design Systems, Inc.

The copied check or assert is added to the cellview. You can view and edit its parameters in the parameter editor pane of the Checks/Asserts assistant.




## Importing Asserts from Text Files Included in the Setup

If you have defined device checks or Spectre asserts in model files, you can include those files.

To import device checks, perform the following steps:

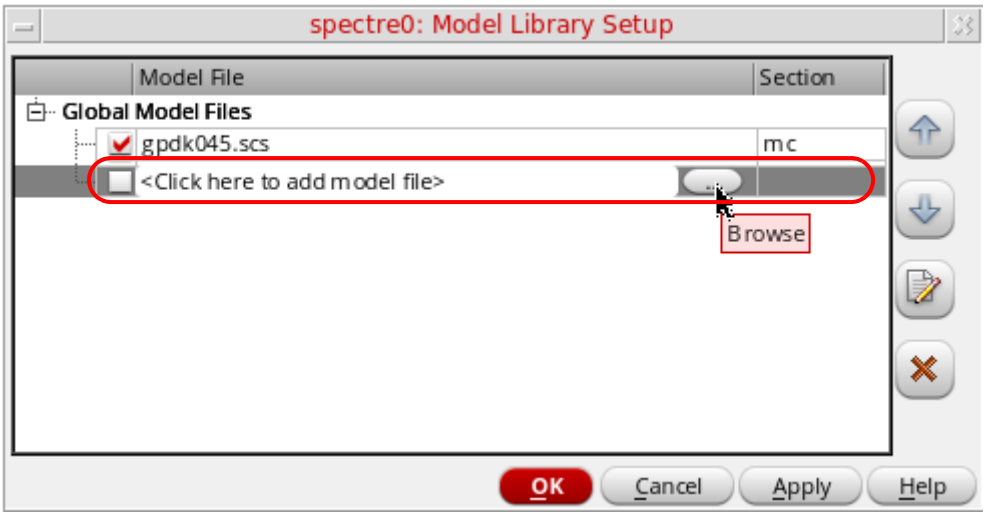

© 2023 Cadence Design Systems, Inc.

1. Right-click the test name and choose *Model Libraries*.

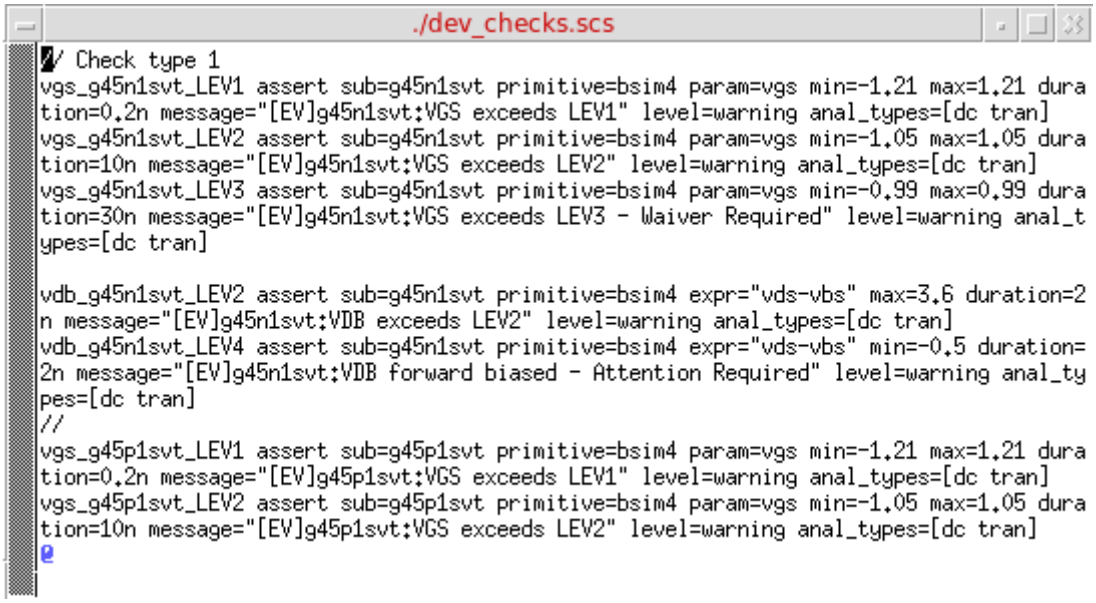
   The Model Library Setup form is displayed.

2. Click `<Click here to add model file>` to view the *Browse* button.

3. Click *Browse* and select a model file that contains the definitions of device checks.




4. (Optional) If you want to view or edit the asserts, select the name of the model file and click *Edit/View Selected Files*.




5. Click *OK* to apply the changes and close the form.


© 2023 Cadence Design Systems, Inc.

The asserts included from the model file will now be included in the netlist and used in simulation runs.

After creating circuit checks, the next steps are to:

- Define violation filters and specifications on the Outputs Setup tab
- Run simulations
- View violations on the Results tab


© 2023 Cadence Design Systems, Inc.

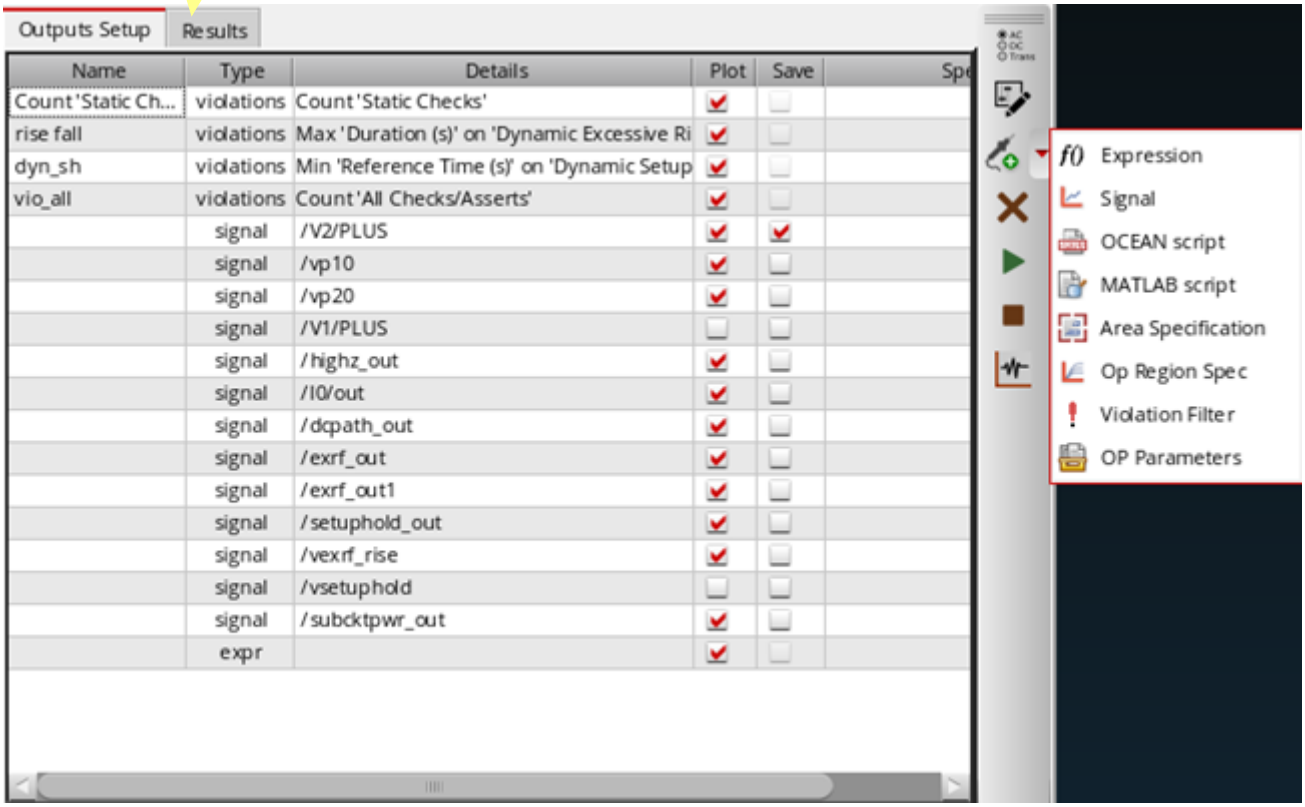
# Filtering Violations

Violation filters provide a way to visualize the different metrics for checks, such as the number, duration, or any user-specified categories of violations. Each violations filter can have its own specification and is treated like any other output.

To create a violation filter, perform the following steps:

1. On the Add Outputs drop-down on the Run toolbar, choose click *Violation Filter* to add a new output to show violation filters for checks or asserts.

***The Outputs Setup tab in ADE Explorer***

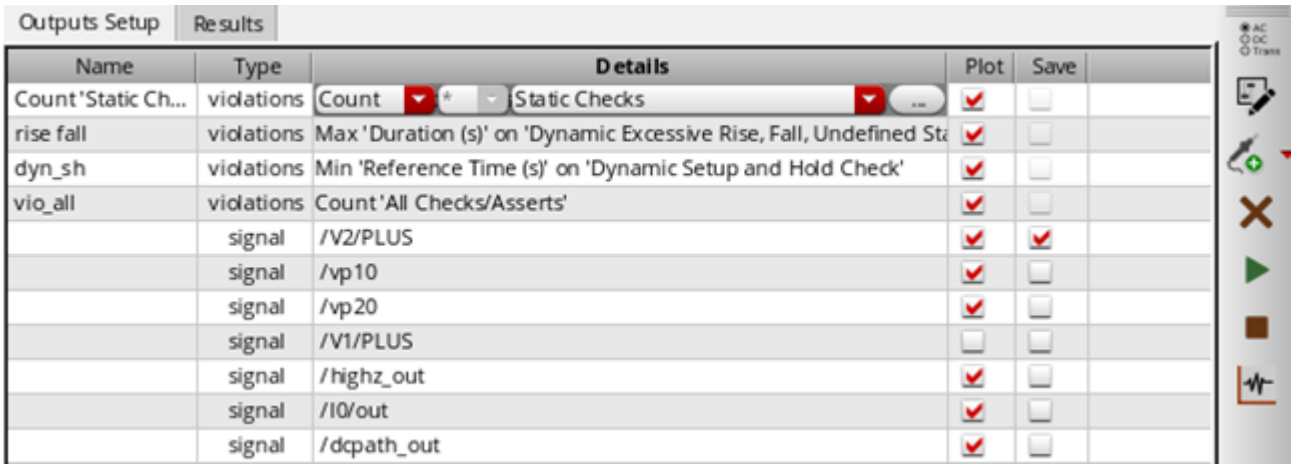



A new row is created with its type set as `violations`.

2. Enter a name in the *Name* column for the violation output.


© 2023 Cadence Design Systems, Inc.

**3.** Double-click the *Details* column for this row.



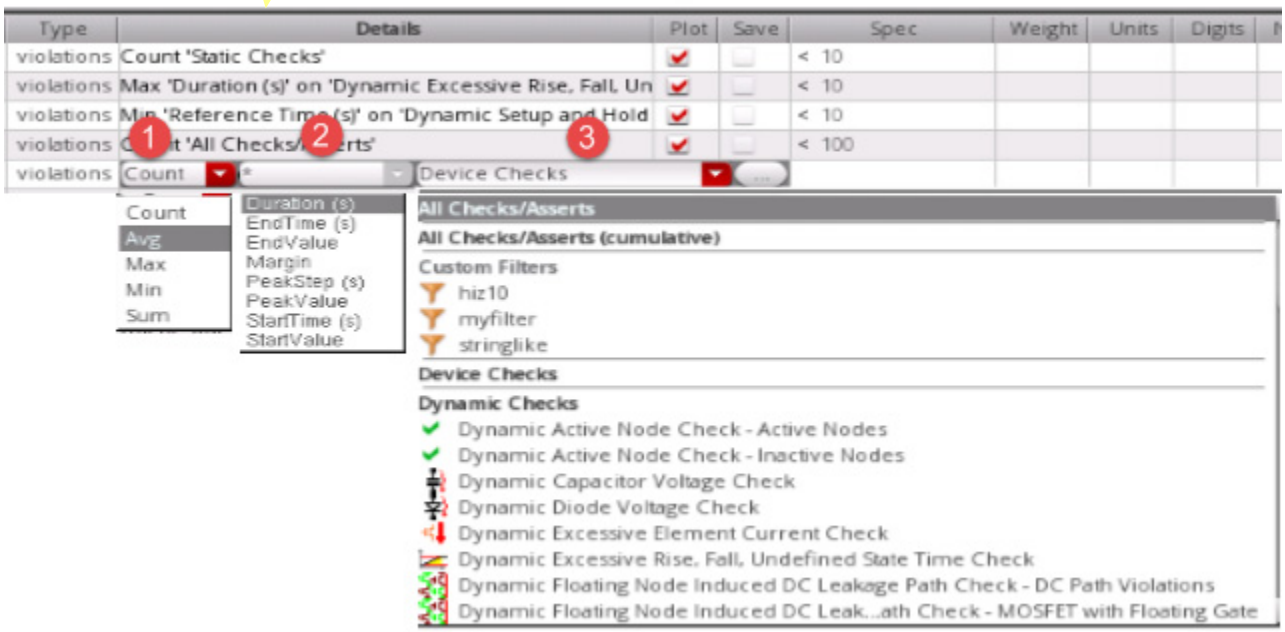

The following three drop-down lists are displayed:

- The first drop-down list is the type of operation you want to perform. For example, you can select `Count` to count the number of violations. Other operations available in this drop-down list are `Avg`, `Min`, `Max`, or `Sum`.
- The second drop-down list specifies the field you want to access within the violation result. For example, for the Avg such as Duration or PeakValue. If the first value is `Count`, the second value defaults to `*`. Otherwise, the choices in the second list vary depending on the item select in the third list.
- The third drop-down list shows the list of checks defined. Choose a check on which you want to run the operation selected in the first drop-down list. You can select an individual violation type, for example, `Dynamic HighZ Node Check` or a grouped


© 2023 Cadence Design Systems, Inc.

category, such as `Dynamic Checks` or `All Checks/Asserts`. The second list of fields is updated depending on the selection in this list.

***The Details column on the Output Setup tab***




4. Select values from the drop-down lists to build an expression for the violation filter.

   Example expressions:

   - `Count 'Device Checks'`
   - `Max 'RMS(W)' on 'Dynamic Subckt Port Power Check - Port Power Report'`

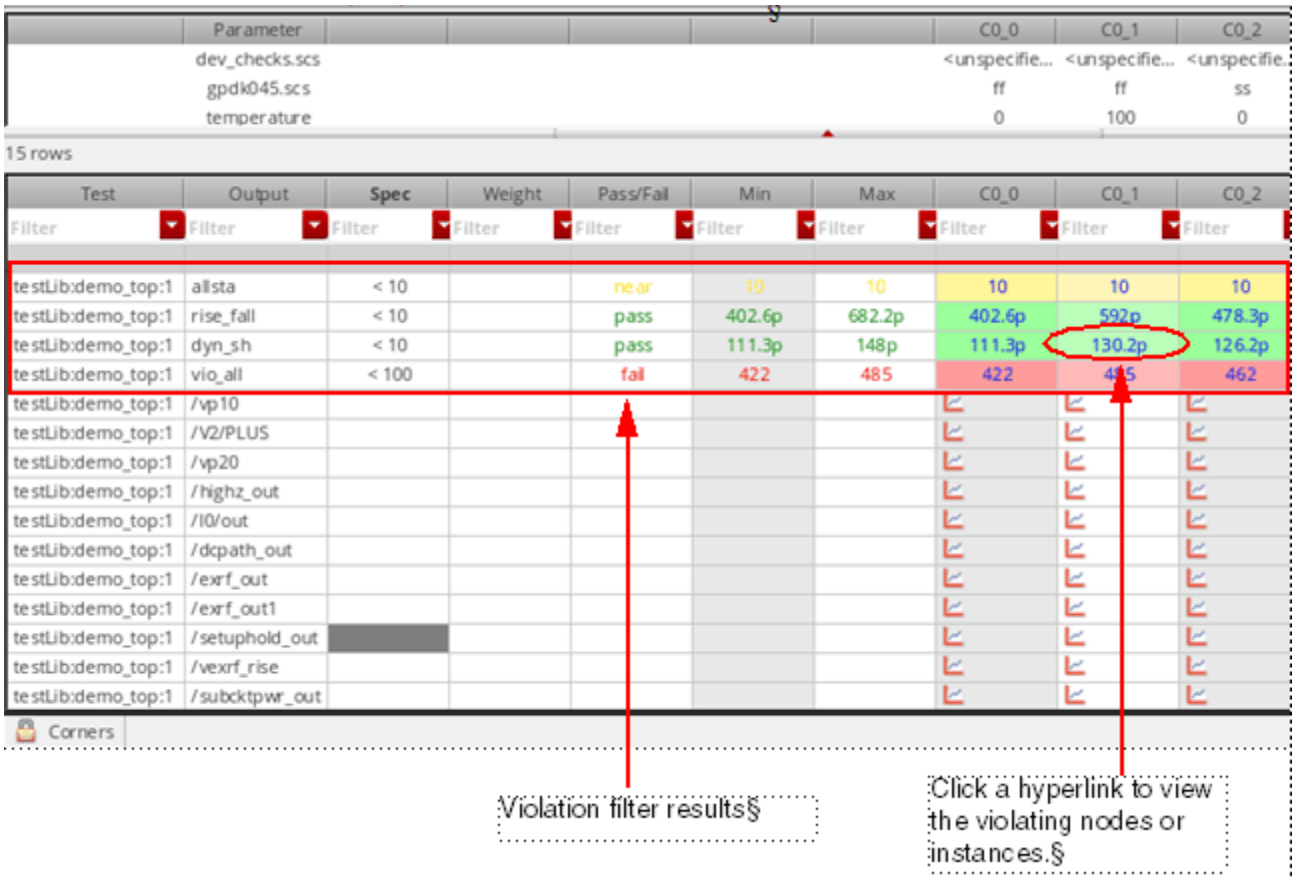
5. Add a specification for the violation in the *Spec* column for the output.


© 2023 Cadence Design Systems, Inc.

# Running Simulations and Viewing Violations

After adding checks in the Checks/Asserts assistant and defining violation filters in the Outputs Setup pane, you can run a simulation to run the checks on the simulation results. The `Detail` results view shows a consolidated report of the violation filter expressions defined in the Output Setup tab. For example, `Count` will report the number of violations of the specified type.

If no violations are found, the result for the violation filter shows `No violations`. When violations are reported, the status is either `pass` or `fail`.




The violation filter output value for each corner is a hyperlink. Click any value with hyperlinks to display the detailed results that contributed to this value. The net- or instance-wise details are opened in `Checks/Asserts` results view. With an aim to resolve violations, you can use the hyperlinks in that detailed view, find the violating nodes or instances, waive violations, and add comments.

The `Checks/Asserts` results view provides a lot of controls to customize the display of violation details.


© 2023 Cadence Design Systems, Inc.

When the results are displayed in the *Results* tab, you can perform the following tasks in the Results tab:

- View itemized violations details
- View cumulative violation details
- View summary of violations
- View instances with violations on schematic
- Create a custom violation filter
- View violation in ViVA XL graphs
- View violation reports in Web browser

For detailed information, refer to Running Simulations and Viewing Violations in *Virtuoso ADE Assembler User Guide*.

## Specifying the Global Device Check Options

You can specify the global device check options, such as start and stop times for a transient analysis and default severity for various Spectre and AMS analyses.

To open the device check options form:

1. Choose *Simulation – Option – Analog*.

   The Simulator Options form appears. For more information about how to use this form, see Setting Spectre Options on page 102

2. Open the *Check* tab.
3. Click the *Options* button available at the bottom of the *Check* tab.


© 2023 Cadence Design Systems, Inc.

The Device Checking Options form appears, as shown in the figure below.

When you open this form, the *Transient* tab is open by default. You can also use this form to specify the global device check options for various analyses.

The following fields are displayed on the *Transient* tab:

- *Start Time for Device Checks*—Specify the beginning time at which device checks are to be enabled or disabled for transient analyses.
- *Stop Time for Device Checks*—Specify the end time at which device checks are to be enabled or disabled for transient analyses.


© 2023 Cadence Design Systems, Inc.

- *Severity*—Select the default severity level for the message that is displayed when Spectre reports violations for the device checks that are enabled for transient analyses. The available options are: `None`, `Error`, `Warning`, `Notice`, and `Fatal`.

  The specified severity level overrides the severity level specified for individual checks in the Checks and Asserts assistant.

  **Note:** If you set the severity to `None`, the severity level specified for individual checks in the Checks and Asserts assistant will be used.

- *Check Windows*—Specify the time windows when device checks will be enabled. In the *Begin* and *End* columns, specify the begin and end times for each time window, one per line in the table.

- *Enable All Checks*—Select this check box to enable all the device checks for transient analyses.

  **Note:** If this check box is not selected, only the device checks that are enabled for transient analyses in the Checks and Asserts assistant are used for transient analyses.

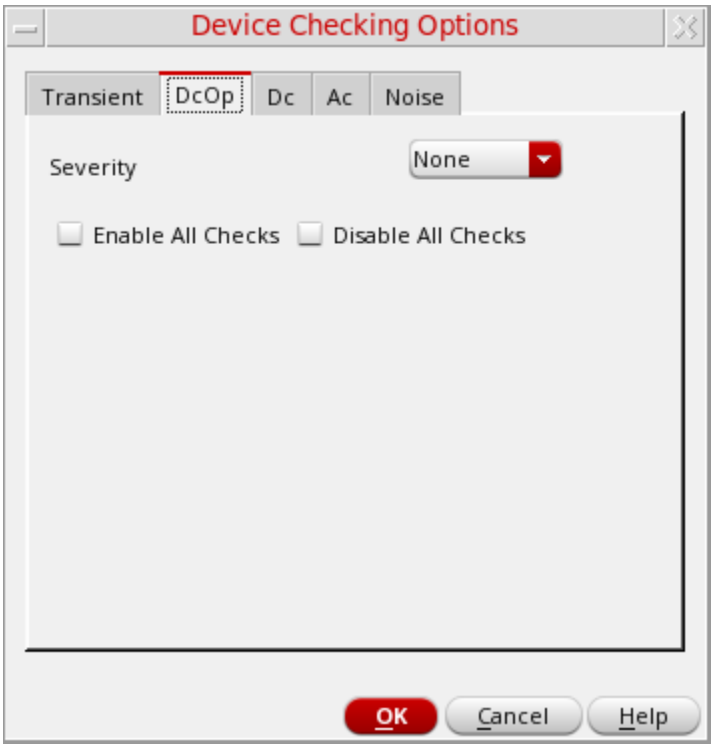
- *Disable All Checks*—Select this check box to disable all the device checks for transient analyses.

- *Additional arguments*—Sets the additional options.

- *Filter*— Select *extreme* to report less assertion violations by omitting less severe repetitions. The available options are `none` and `extreme`.


© 2023 Cadence Design Systems, Inc.

The remaining tabs, *DcOp*, *Dc*, *Ac*, *Sp*, *and Noise,* include only three fields as shown in the figure below:





© 2023 Cadence Design Systems, Inc.

© 2023 Cadence Design Systems, Inc.

# 14

# Helping Simulations to Converge

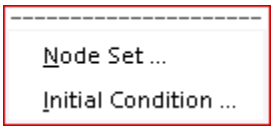
This chapter describes how you can help a troublesome simulation to converge. Select topics from the following list to view more information.


- Commands for Forcing Convergence on page 989
- Selecting Nodes and Setting their Values on page 992
- Loading an Existing Convergence Setup on page 995
- Highlighting Set Nodes on page 996
- Storing a Solution on page 996
- Restoring a Solution for Spectre on page 997


## Commands for Forcing Convergence

You use the commands in the *Simulation – Convergence Aids* menu to help the simulator find a solution when it fails to achieve convergence. Once you get the simulation to converge, you can save the DC and Transient solutions. When you resimulate, you can save time by restoring the saved solutions.

There are two commands to help the simulator find a solution:




- *Node Set*, which provides an initial guess for nodes in any DC analysis or the initial condition calculation for the transient analysis
- *Initial Condition*, which provides initial conditions for nodes in the transient analysis

**Note:** An additional command, *Force Node*, appears when HSpiceD simulator is selected. See Force Node for more information.


© 2023 Cadence Design Systems, Inc.

Refer to the simulator manual for specific details about the commands that help circuits converge. Not all simulators support these three commands, and simulators implement these methods differently.

## Node Set

To set an initial DC voltage on selected nodes, use the *Simulation – Convergence Aids – Node Set* command.

For Spectre, the node set is used to provide an initial guess for nodes in any DC analysis or the initial condition calculation for the transient analysis. It netlists to

```
nodeset node=value
```

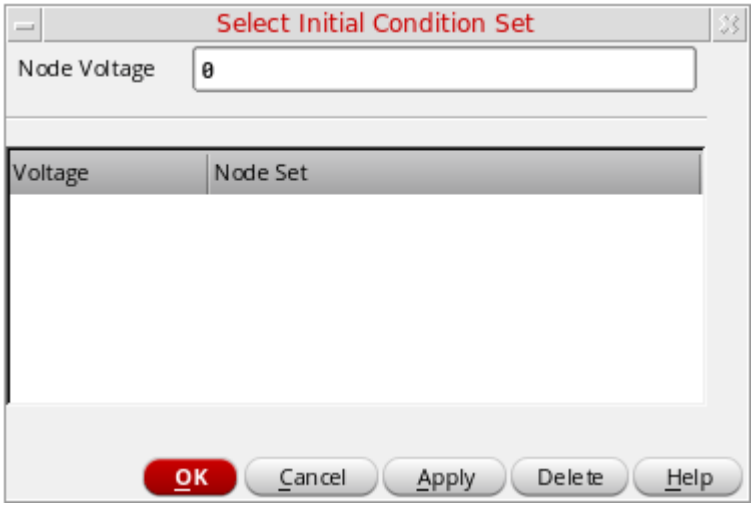
For more information, see the *Node Sets (nodeset)* section in the *Other Simulation Topics* chapter of the *Spectre Circuit Simulator Reference*.

For other simulators, the *Node Set* command is equivalent to

```
.NODESET v(node)=value
```

## Initial Conditions

To set an initial transient voltage on selected nodes, use the *Simulation – Convergence Aids – Initial Condition* command. The Select Initial Condition Set form appears with the schematic design.




For Spectre, initial conditions are used to provide initial conditions for nodes in the transient analysis. Initial conditions are accepted only for inductor currents and node voltages where the nodes have a path of capacitors to ground. This is netlisted to


© 2023 Cadence Design Systems, Inc.

`ic node=value`

For more information, see the Initial Conditions (ic) section in the *Other Simulation Topics* chapter of the *Spectre Circuit Simulator Reference*.

For other simulators, the *Initial Condition* command is equivalent to

`.IC node=value`

**Note:** In case the selected object is an instance, then the mapped instance name should be followed by the terminal index number of the instance in the *Initial Condition* statement.

`ic node:index_number=value`

## Force Node

Force Node command sets the voltage on a node and locks it at that voltage during the entire simulation. To set a node to a specific voltage throughout the simulation, use the *Simulation – Convergence Aids – Force Node* command.

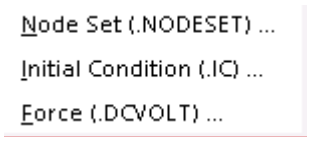
**Note:** This command appears only when HSpiceD simulator is selected.

For details on this command, see the reference manual for the simulator that you use.

One way to use this feature is to store the DC solution from a simulation with *Force Node* active, remove the *Force Node* setting, restore the DC solution, and run another simulation.

## HspiceD Convergence Aids

The *Convergence Aids* submenu appears and works slightly different.




- *Convergence Aids – Node Set (.NODESET)* initializes specified nodal voltages for a DC operating point analysis. The `.NODESET` statement is used to correct convergence problems in a DC analysis. Setting nodes in the circuit to values that are close to the DC operating point solution enhances the convergence of the simulation. The *Select Node Set* form works in the same way as the Spectre Direct interface. The netlist will contain the `.NODESET` statement line.
- *Convergence Aids – Initial Condition (.IC*) or *Convergence Aids– Force (.DCVOLT)* sets the transient initial conditions. The initialization depends on whether the UIC parameter is included in the .TRAN analysis statement. If the UIC parameter is


© 2023 Cadence Design Systems, Inc.

specified in the .TRAN statement, the HSPICE simulator does not calculate the initial DC operating point. Consequently, the transient analysis is entered directly.

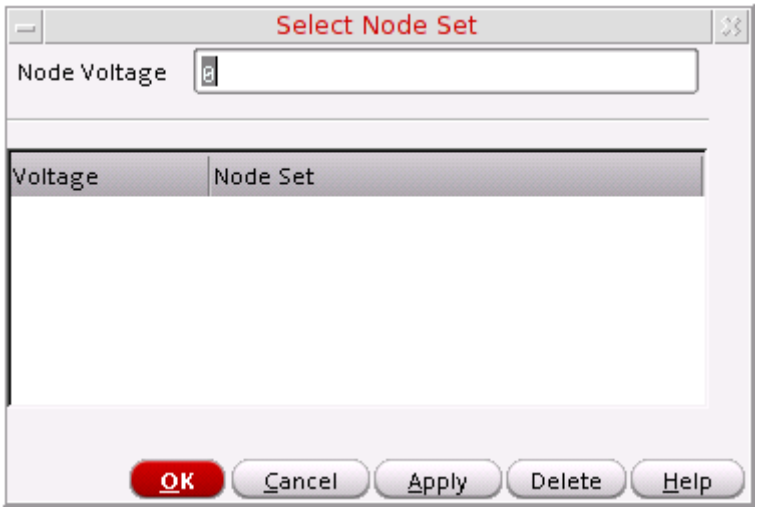
- *Select Initial Condition Set* and the *Select Force Node Set* work in the same way as the Spectre Direct interface. The netlist contains the `.IC` and the `.DCVOLT` statement line, whichever the case may be.

# Selecting Nodes and Setting their Values

Perform the following steps to select a node and set its voltage:

1. Choose *Simulation – Convergence Aids* and a convergence command (*Node Set, Initial Condition,* or *Force Node*).

   The schematic opens in a new tab and a form appears to enter the voltage. Each command displays a different form. The *Select Node Set* form is shown below.

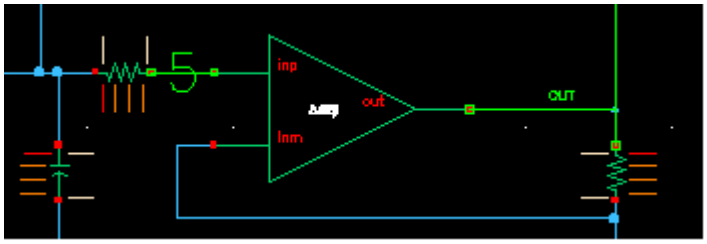



2. Type the voltage.
3. Click in the schematic to select the first node.

   The node name appears in the form.


© 2023 Cadence Design Systems, Inc.

In the schematic, the node is highlighted and the voltage appears. For split nets, the system labels only the driving cell.




4. To set other nodes to the same voltage, select them.
5. To set other nodes to a different voltage, change the voltage, and select other nodes.
6. Click *OK* or *Cancel* when you are finished selecting nodes.

**Note:** You can also set the node voltage in the *Select Node Set* form using a VAR("`myVoltage`") variable expression syntax. The variable, `myVoltage`, will then appear in the *Design Variables* section in the Setup assistant of Explorer. To provide a value for the variable, right-click the variable and choose *Edit Variable.* The *Editing Design Variables form* appears. Specify the value in the *Value* column of the *Design Variables* table. If you provide an expression, the value of the expression is calculated when you create the netlist.

You can select nodes at any level of the hierarchy. When you select an interface node at a lower level, the node is highlighted, but the voltage value appears only on the higher-level schematic.

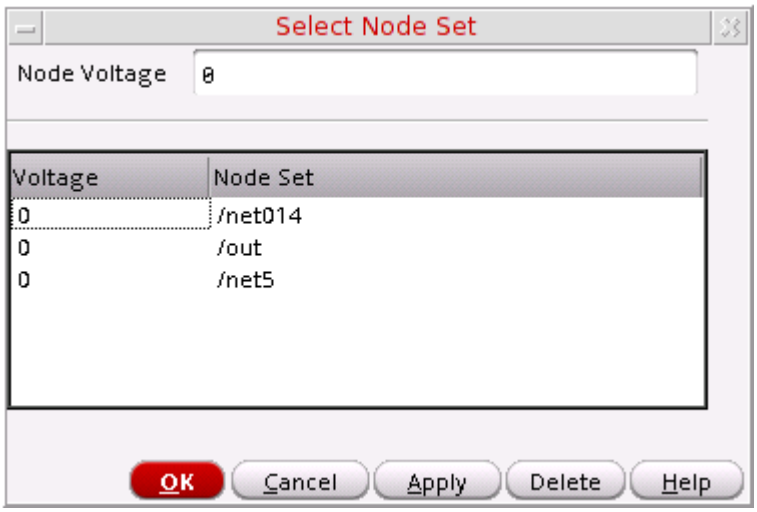
## Releasing Voltages

Perform the following steps to release the node set, initial condition, or force node voltage settings:

1. Choose *Simulation – Convergence Aids* and a convergence command (*Node Set*, *Initial Condition*, or *Force Node*).


© 2023 Cadence Design Systems, Inc.

The schematic opens in a new tab and a form appears listing all of the nodes that have been set. The *Select Node Set* form is shown below.




2. Click the net in the schematic to release it, or click the net name in the form and click *Delete*.

   On the form, you can select several nodes to release by holding down the left mouse button and dragging through the list box.

3. Click *OK* or *Cancel* when you are finished releasing nodes.

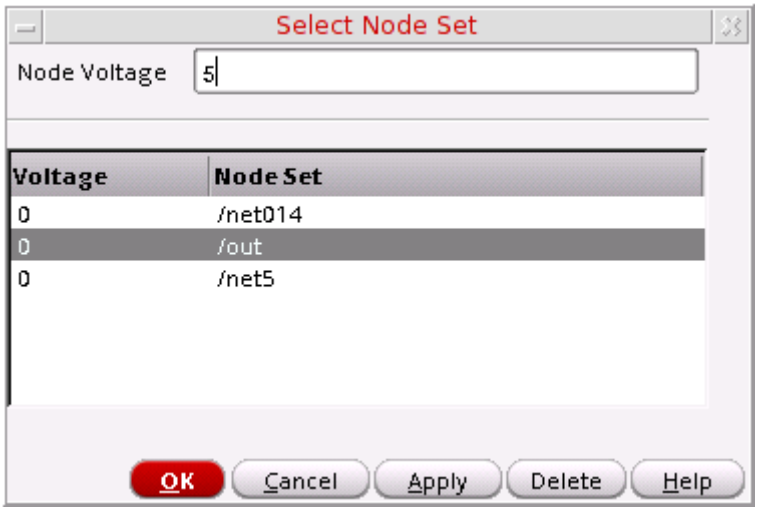
## Changing Voltages

Perform the following steps to change the value of the voltage set on a node:

1. Choose *Simulation – Convergence Aids* and a convergence command (*Node Set*, *Initial Condition*, or *Force Node*).


© 2023 Cadence Design Systems, Inc.

The schematic opens in a new tab and a form appears listing all of the nodes that have been set. The *Select Node Set* form is shown below.




2. Click the node name to highlight it.
3. Type the new voltage value, and click *Apply*.
4. Click *OK* or *Cancel* when you are finished changing voltages.

# Loading an Existing Convergence Setup

Perform the following steps to load an existing convergence setup:

1. Choose *Session – Load ADE L State*.

   The *Loading State* form appears. The form displays the state files in the run directory identified by the *Cell* and *Simulator* fields.

2. Choose a run directory with the *Cell* and *Simulator* fields.

   The list box shows the saved states for the cell and simulator combination.

3. Click a *State Name*.
4. Check that *Convergence Setup* check box is selected, and click *OK*.


© 2023 Cadence Design Systems, Inc.

# Highlighting Set Nodes

While the *Select* commands are active, the system highlights selected nodes and labels them with the voltages you set. After you close the form, the system removes the highlighting and the labels.

Perform the following step to redisplay the highlighting and labels:

➤ Choose *Simulation – Convergence Aids* and a convergence command (*Node Set*, *Initial Condition*, or *Force Node*).

Perform the following step to remove the highlighting and labels:

➤ Close the active *Select* form.

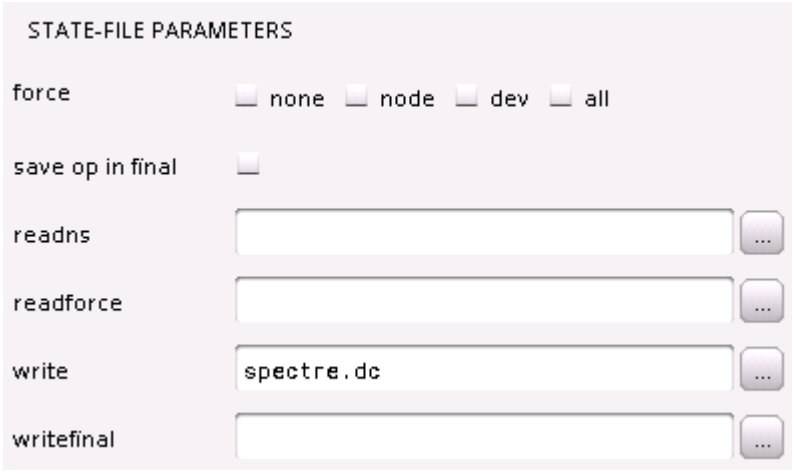
# Storing a Solution

To store a solution for the Spectre simulator, use the *write* and *writefinal* fields in the *State File Parameters* section of any *Analysis Options* form.

This causes the Spectre simulator to write out the initial and final conditions of a transient analysis. By default, the conditions are written to the files `spectre.ic` and `spectre.fc` in the netlist directory.


© 2023 Cadence Design Systems, Inc.

To store the solution of an analysis, use the *write* and *writefinal* fields of the *DC Options* form for the analysis.




**Note:** This does not apply to spectre. Use the analysis options instead.

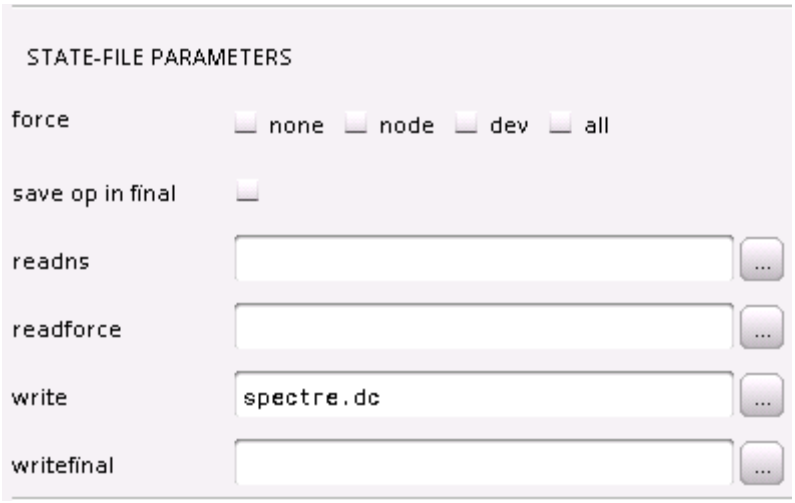
# Restoring a Solution for Spectre

To restore a saved transient solution, use the *readns* field in the *Convergence Parameters* section of the *Transient Options* form. Use the name of the file that was previously used in the *write* or *writefinal* sections of the same form.

To stop using the solution, clear the field.


© 2023 Cadence Design Systems, Inc.

To restore a saved DC solution, use the *readns* field in the *State-File Parameters* section of the *DC Options* form. Use the name of the file that was previously used in the *write* or *writefinal* sections of the same form.




To stop using the solution, clear the field.


© 2023 Cadence Design Systems, Inc.

# 15

# Working with AMS Simulator

ADE Explorer and ADE Assembler provide seamless integration of the Spectre AMS Designer and Xcelium Mixed-Signal simulator. The integration of the Spectre AMS Designer simulator with ADE Explorer or ADE Assembler creates a design environment with the look and feel expected by the analog and mixed-signal designers who already use these tools. When using this integration, you can access designs using the same tools you currently use for pure analog and mixed-signal designs.

You can use the Virtuoso® Hierarchy Editor to specify the cellviews that you want to use in your design. A configuration (config) is a set of binding rules that defines which cellviews are part of the design for a given purpose (such as netlisting and simulation). Using the Virtuoso Hierarchy Editor, you can view the hierarchy of these cellviews and apply view switching to mix and match abstraction levels depending upon which phase of the design cycle you are performing.

To simulate your design with AMS Designer in ADE Explorer or ADE Assembler, you must specify a top-level config cellview for your design hierarchy. The design can contain other config cellviews at lower levels in the hierarchy. See Creating a Configuration Cellview for information on how to create a config view

**Note:** You can use the variable `TESTBENCH_TOPCELL` to access the top cell name of a testbench and to backannotate the SDF delays. This variable is automatically saved in the `runSimulation` script generated by AMS Designer.

For example:

The top cell name is written into the `TESTBENCH_TOPCELL` variable through the `runSimulation` script as follows.

```
export TESTBENCH_TOPCELL="test_sim"
xrun -f xrunArgs
```

The following `xrun` option is specified with a file that uses this variable. On running a simulation, it backannotates the SDF delays and writes them to the log file `sdf.log`.

```
-sdf_cmd_file c_sdf


cat c_sdf
```


© 2023 Cadence Design Systems, Inc.

```
COMPILED_SDF_FILE = "${CDIR}/top_cell.sdf.X",
SCOPE = "${TESTBENCH_TOPCELL}.I0",
LOG_FILE = "sdf.log"
```

**Note:** When ADE Explorer is running a single-point simulation, you can use the 'Optimize Single Point Run' option on the job policy setting form to optimize the run. For more details, refer to the section Optimizing a Single Point Run.

This chapter covers the following topics:

- Creating a Configuration Cellview on page 1001
- ADE Explorer and AMS Integration on page 1010
- Setting up AMS Options on page 1012
- About the AMS Unified Netlister on page 1071
- Setting Up the AMS Netlister Options on page 1112
- Using the runams Command on page 1137
- Viewing the Output Log for AMS on page 1159
- Viewing the Error Explanation for AMS on page 1160
- Using the SimVision Debugger on page 1161

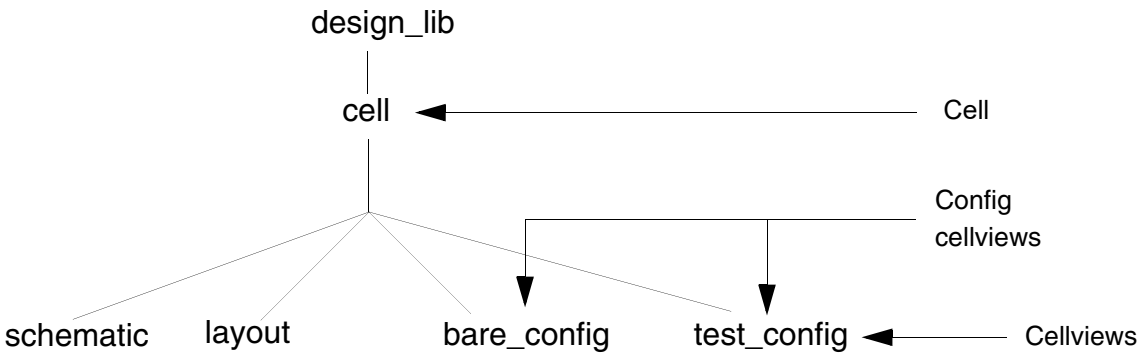
Video

For an overview of the above steps, see the Running AMS Simulation in ADE Explorer video on Cadence Online Support.


© 2023 Cadence Design Systems, Inc.

# Creating a Configuration Cellview

A configuration (or config) is a view of the cell (or cellview). You can have different config cellviews for different purposes.




To simulate your design in the AMS Designer Virtuoso ADE Explorer environment, you must have a top-level config cellview. The top-level config cellview can be a HDL text module or a schematic. The top-level config can contain other config cellviews lower in the hierarchy.

Configurations let you bind to different cellviews as your design evolves from concept to finish. For example, you might begin the design process using high-level behavioral models of your design components; later, you might insert modules into test fixtures; you might replace behavioral descriptions with schematics and finally, add post-layout views, as the design process approaches implementation.

You can create configuration rules that define what views to include in the hierarchy at three different levels:

| | |
|---|---|
| Global level | Using a global view list and stop list |
| Cell level | Using cell-based view lists — which affect the cell as well as the structures below the cell in the hierarchy — and cell bindings |
| Instance level | Using instance-based view lists — which affect the instance as well as structures lower in the hierarchy — and instance bindings. |

You can use the Virtuoso® Hierarchy Editor to create a configuration (config) cellview as follows:

1. Choose *File — New Config* to view the *New Configuration* form.


© 2023 Cadence Design Systems, Inc.

2. On the *New Configuration* form, click *Use Template*.

   The *Use Template* form appears.

3. Select *AMS* from the *Name* drop-down list.

   The resulting view list is:

   ```
   verilogams veriloga behavioral functional schematic symbol
   ```

   **Note:** Symbol views must have associated underlying models that describe the represented device for AMS simulation.

4. Click *OK*.

   All design instances and their cell bindings appear in the Virtuoso Hierarchy Editor window.

5. (Optional) In the *Global Bindings* group box, you can type directly in the fields to edit the lists as required. For example, VHDL users might want to add a wildcard asterisk to View List.

6. Choose *View - Update* to check and save the new configuration.

   An *Update* prompt appears.

7. Click *OK* to save the config view.

## Setting Up the Default Simulator for a Configuration

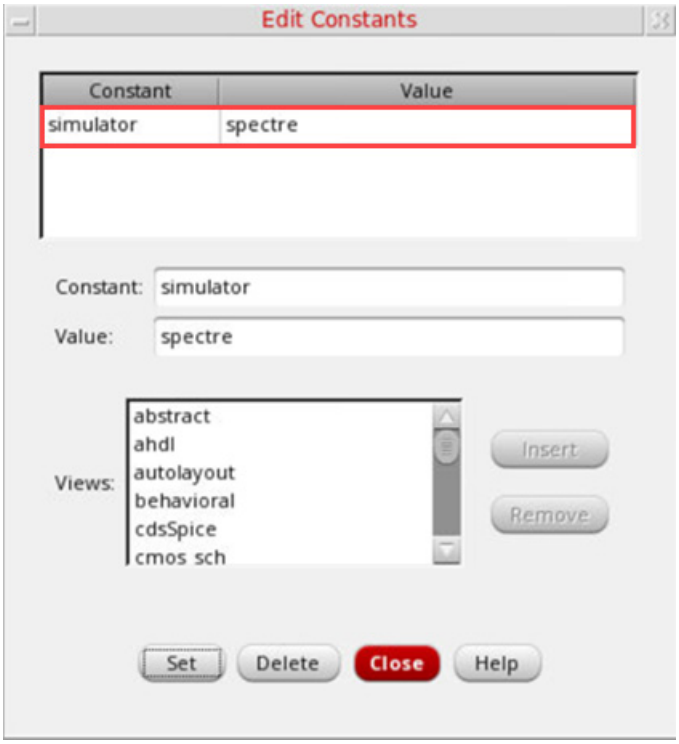
By default, a configuration uses the default simulator which is set using the variable `asimenv.startup simulator`.

To use a specific simulator for a particular configuration,

1. Set the variable `hed.ade setAmsSimulator` to `nil`.
2. Define a constant named `simulator` through the Edit Constants form in Virtuoso Hierarchy Editor.
3. Specify a simulator name as the constant value.


© 2023 Cadence Design Systems, Inc.

For example, setting the value `spectre` for the constant `simulator`, will set Spectre as the default simulator for the config view.




Here, the value of the `simulator` constant will override the default simulator, set using `asimenv.startup simulator`. If the simulator constant is invalid, an error message is displayed in the CIW and the default simulator, Spectre, is used.

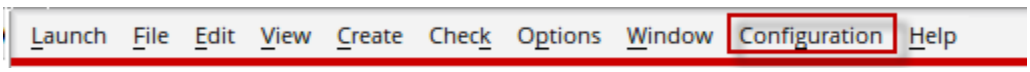
For more information about adding constants, see Using Constants.

## Setting Up a Config Cellview Using Virtuoso Schematic Editor or Virtuoso Text Editor

While working on a schematic cellview or a text cellview, you can set up a corresponding config cellview directly from Virtuoso Schematic Editor (VSE) or Virtuoso Text Editor.

To view the *Configuration* menu in VSE and Text Editor,

➡ Open VSE or Text Editor and choose *Launch – Plugins – Mixed Signal Options – Configuration.*



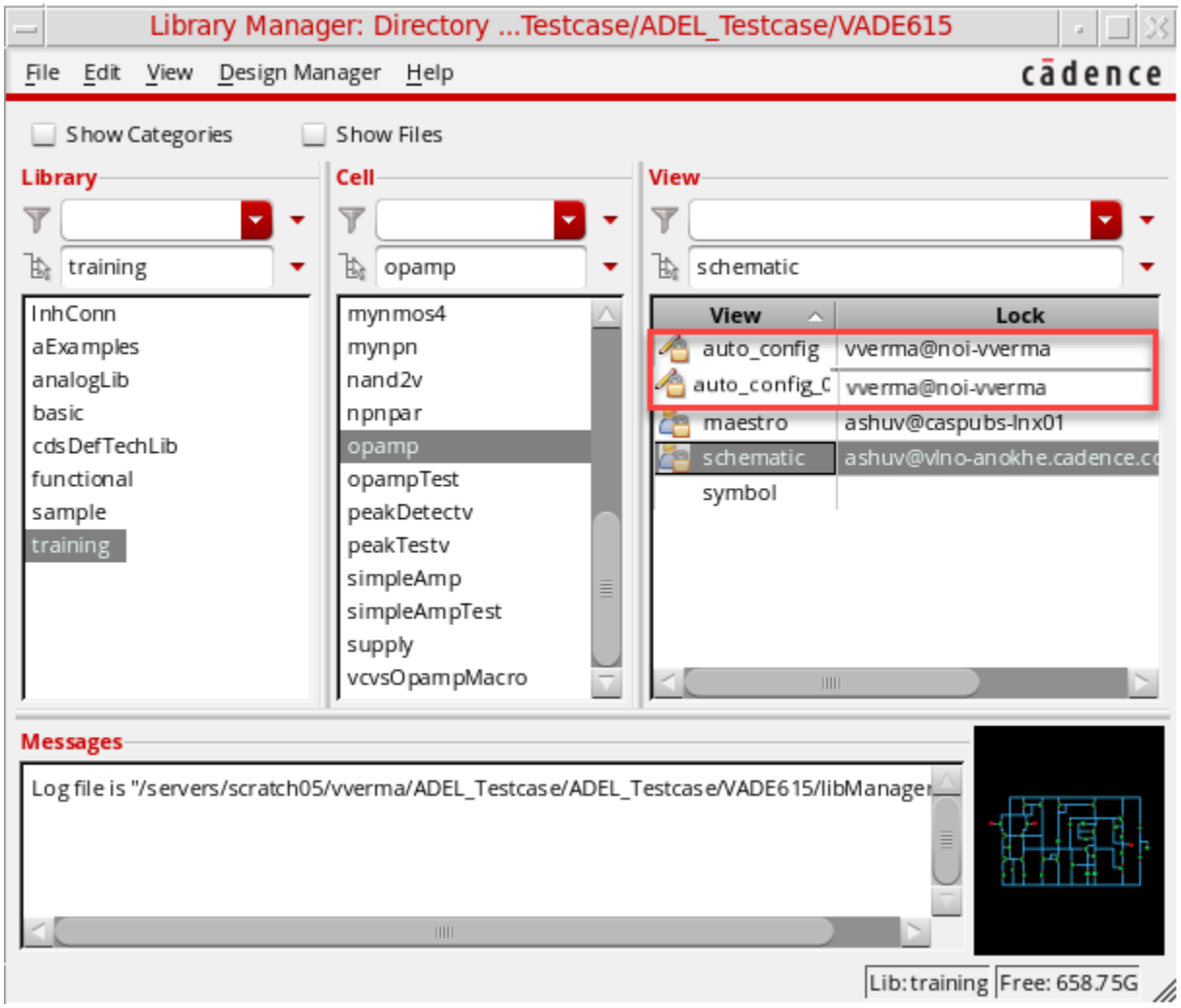


© 2023 Cadence Design Systems, Inc.

The *Configuration* menu includes the following options:

- *Auto Create Configuration*—Creates a new config cellview automatically. By default, the name of this new config cellview is `auto_config`. However, you can specify a different name for this cellview using the autoConfigNameForConfigCreation environment variable.

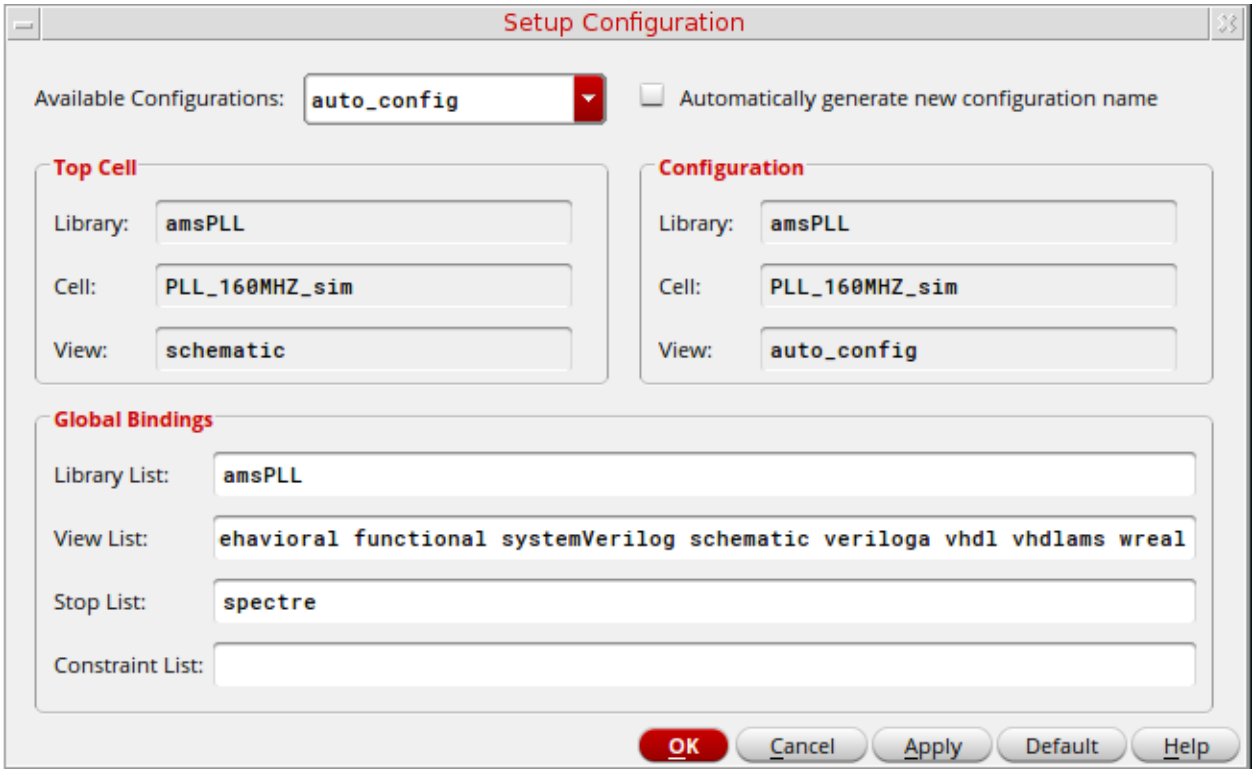
  **Note:** If you try to create a new cellview automatically when the cellview `auto_config` already exists, it creates a new cellview `auto_config_0`. These `auto_config` cellviews are created incrementally. The added cellviews can be seen in the View column of Library Manager.





© 2023 Cadence Design Systems, Inc.

- *Setup Configuration*–Sets up a config cellview. When you select this option, the Setup Configuration form appears as shown in the figure below:




In this form, you can create a new config cellview and specify global bindings options for this new cellview. If the config cellview already exists, you can use this form to modify these options.

Specify the following form fields to set up a config cellview. Depending upon whether a config cellview already exists or not, perform the following steps:

**a.** In the *Available Configurations* drop-down list, select a config cellview.

- If a config cellview does not exist, the *Available Configurations* field still displays a config cellview name, `auto_config`, and a new config with this name is created when you click *OK*.
- If a config cellview exists and you want to create a new config cellview using the settings from the existing cellview, select the *Automatically generate new configuration name* check box. The config cellview is created with the specified name when you click *OK*.
- If a config cellview exists and you do not select the *Automatically generate new configuration name* check box, the selected config cellview is modified when you click *OK*.


© 2023 Cadence Design Systems, Inc.

**b.** The *Top Cell* section displays the name of the library, cell, and cellview used as the top cellview of your design.

**c.** The *Configuration* section displays the name of the library and cell in which the selected config cellview is to be placed, along with the name of the config cellview

**d.** In the *Global Bindings* section, specify the following options:

- *Library List* – Specify the libraries for cells that do not have fixed library bindings. List the libraries in the order you want them to be searched.
- *View List* – Specify the views you want to open/display (something is missing here) in your configuration. The view list applies to every level of the configuration and determines which view is selected for each object in the design, unless overridden by a cell or instance binding. You can use the asterisk character (*) as a wildcard in the *View List* field. The default view list is determined by the value of the viewListForConfigCreationenvironment variable.
- *Stop List* – Specify a list of views that are to be treated as leaf nodes, that is, not to be expanded. This field is optional. The default stop list is determined by the value of the stopListForConfigCreationenvironment variable.

**Note:** You can use constants in the *View List* and *Stop List* fields. A constant is a symbolic name used to represent a set of views. For information about creating and using constants, see Using Constants in the *Virtuoso Hierarchy Editor User Guide*.

- *Constraint List* – Specifies a list of constraint views that apply to the design. This field is optional. The default constraint list is determined by the value of the constraintListForConfigCreation environment variable.

List the entries in each list in order of preference; separate entries with spaces.

**e.** Click *OK*.

The config cellview with the specified name and settings is added in Library Manager. You can double-click the cellview to open and edit the configuration in Virtuoso Hierarchy Editor.

For more information, see:

- Editing Configurations in the *Virtuoso Hierarchy Editor User Guide.*
- Saving Configurations in the *Virtuoso Hierarchy Editor User Guide.*


© 2023 Cadence Design Systems, Inc.

### Using VHDL Design Units in a Configuration

To instantiate a VHDL design unit in a text module or schematic, bind to the `architecture` view of the VHDL design unit in the configuration. You can also instantiate Verilog-AMS modules in VHDL modules.

### Netlisting to Make the HDL Design Unit Information Current

If you edit an HDL design unit through Virtuoso (such as a SystemVerilog text module ) using the text editor (such as vi), you must netlist your design to ensure that the Virtuoso® Hierarchy Editor has up-to-date information. Otherwise, design expansion might not result in what you expect.

The integration of the AMS Designer simulator and Virtuoso ADE Explorer has the following features:

- Connect Rules

  You can point to existing connect rules or create your own by parameterizing existing rules. The form allows you to work with multiple connect rules and auto-compiles all the built in or modified rules.

- MATLAB/Simulink.

  The ability to run a co-simulation using MATLAB/Simulink with AMS is now available in ADE Explorer. You can start MATLAB® before AMS starts by setting specified waiting time and run the cosimulation with general analog simulation flow in Explorer. You can also start MATLAB independently and run the cosimulation. For more information, see Using MATLAB/Simulink on page 1066

- Global Signals

  You can use the Global Signals form to declare a signal that is used as an out-of-module signal reference. For more information, see Working with Global Signals in AMS on page 1026.

- SimVision Integration

  You can run simulations interactively using the SimVision debugger, by changing the run mode to interactive. Cross-probing from schematics works with the SimVision integration. For more information, see Using the SimVision Debugger on page 1161.

- LogFile Utility

  You can look at all the individual log files, opened in an xterm window, or you can use the XMBrowse logfile utility that matches a particular error in a logfile back to the original source. For more information, see Viewing the Output Log for AMS on page 1159.


© 2023 Cadence Design Systems, Inc.

- Error Explanation

  You can view detailed explanation of the error for AMS in the Error Explanation form. To view an error, you need to enter the error string. However, the errors displayed for AMS are the ones that are present in the log files that are created in the psf directory while a session is being run. For more information, see Viewing the Error Explanation for AMS on page 1160.

- Default Disciplines

  You can specify disciplines on a library, cell, cell terminal, instance, instance terminal and net from the Composer UI. You can autocreate a discrete discipline and Explorer auto-compiles the discipline for you. For more information, see Default Digital Discipline Selection on page 1169.

- Advanced Analyses

  Advanced Analyses such as parametric analysis works with AMS.

- Available Analyses

  Transient, AC, DC, Noise, and Envelope analyses are available when you use the AMS simulator with Explorer. You can save the DC operating point.

- Outputs

  Use the Explorer *Output* options to *Save All Signals* or to *Select Specific Signals*. You cannot save AC currents.

- Single Button *Netlist and Run*

  Netlist files must be compiled and elaborated before they can be simulated. When you press *Netlist and Run* a sequence of tools is invoked including the simulator.

  - Schematics are translated to Verilog-AMS netlists
  - Netlists are compiled
  - Elaboration runs
  - Simulation runs

  Each tool produces a separate log file. When any tool fails, the CIW displays a failure message. Look in the tool's log file for descriptive error messages. For details, refer to the section Viewing the Output Log for AMS on page 1159.

- Full Support for Virtuoso ADE Explorer Display Tools

  The integration of the AMS Designer simulator and ADE Explorer makes the full set of available tools. In particular, you can examine waveforms with the Virtuoso Visualization


© 2023 Cadence Design Systems, Inc.

and Analysis XL tool and take advantage of their superior performance for large mixed-signal designs as well as analog-centric capabilities. You can also bring up SimVision as a simple waveform tool after simulation as well. You can backannotate DC and transient operating point information to the schematic. You can use data access features, such as Calculator, Results Browser, and the Direct Plot form.

- OCEAN

  Ocean provides full support for the AMS Designer simulator including several new commands such as `connectRules()`. For details of the commands, refer to *OCEAN Reference*.

- Distributed and Remote Simulations

  Use network mode for distributed AMS simulations.

- Visual Display of Signal Domains

  In the schematic window, you can highlight analog nets and digital nets in different colors, with indicators showing the location of automatically inserted connect modules. Visualization of a mixed-signal design can help you tune the design for desired characteristics.

- Display Partition

  The Display Partition capability of the AMS environment and simulator in Virtuoso ADE Explorer is similar to the display partition capability used in verimix.

To read some frequently asked questions about AMS-in-ADE Explorer, choose *Session – FAQ* from the ADE Explorer window.


© 2023 Cadence Design Systems, Inc.

# ADE Explorer and AMS Integration

Following significant differences are introduced with the AMS integration:

- AMS brings a new and complete Unified Netlisting (UNL) solution to Virtuoso ADE Explorer. The AMS UNL flow adds increased speed, flexibility, and capacity to Virtuoso ADE Explorer and maintains the netlisting compatibility between Explorer and AMS. For more information about the AMS UNL flow, see About the AMS Unified Netlister on page 1071.
- Both the *Netlist* command and the *Netlist and Run* command can potentially call several tools to:
  - Compile updated text views
  - Netlist updated configurations and cellviews
  - As necessary, call several additional tools
    - *xmvlog* or *xmvhdl* to compile
    - *xmelab* to elaborate
    - *xmsim* to simulate

  Messages in the CIW indicate which tool is running. Each tool writes its own log file.
- You cannot directly use the following in designs:
  - Parameters declared in model files. This is due to language boundary issues in Spice and Spectre.
  - Parameters defined in definition files. This is due to language boundary issues in VerilogAMS.
- You can specify a text block as the top level in your configuration, as follows:

  **a.** Compile your top-level module manually, for example, by running this command:

  ```
  cdsTextTo5x -LIB lib1 -SYMBOL symbol textTop.v
  ```

  After you run this command:
  - A new library, `lib1`, is created if it does not exist.
  - A 5x structure for cellview `lib1.textTop:verilog` is created.
  - A new cellview `lib1/textTop/symbol` is created.

  **b.** In the Hierarchy Editor, create a config view with this text, `textTop.v`, as top.


© 2023 Cadence Design Systems, Inc.

**c.** Open an ADE Explorer session with the newly created config view.

### cds_alias

cds_alias is a simple cell which is defined in the library `"basic"`. This cell is required only if your design contains the cds_alias instances. If the design that you are using contains `cds_alias` and you have not defined a library "basic" in your cds.lib, then a default cds_alias cell is created under the top design.

For example, if the cell, cds_alias is in library `"basic"`, you need not do anything. It will be compiled under implicit_tmp_dir/basic/cds_alias/functional. However, if the library `"basic"` is not present in cds.lib, AMS-ADE Explorer will itself compile it in `implicit_tmp_dir/<design_lib>/cds_alias/functional`.


© 2023 Cadence Design Systems, Inc.

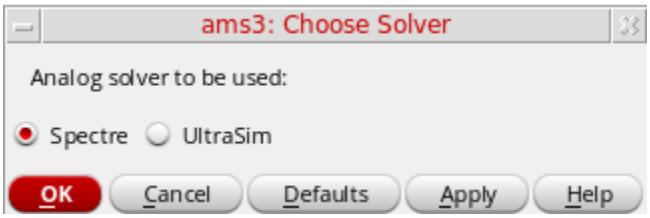
# Setting up AMS Options

By default, when you select *ams* as your simulator, Spectre is selected as the solver for the simulation. But you can select the UltraSim solver by clicking the *Simulation - Solver* menu option, and selecting *UltraSim* from the Choose Solver form. After selecting the solver, you can set Spectre AMS Designer and Xcelium Mixed-Signal simulator options by choosing the appropriate option from the *Simulation – Options* submenu.

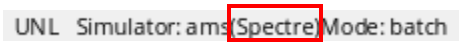
**Note:** Starting Xcelium 20.03, the AMS simulator does not support the UltraSim solver.

### Choosing a Solver

Choose *Simulation – Solver* to open the Choose Solver form, in which you can select either *Spectre or UltraSim* as the solver.




Your choice appears next to the name of the selected simulator below the title bar as highlighted in the snapshot below.



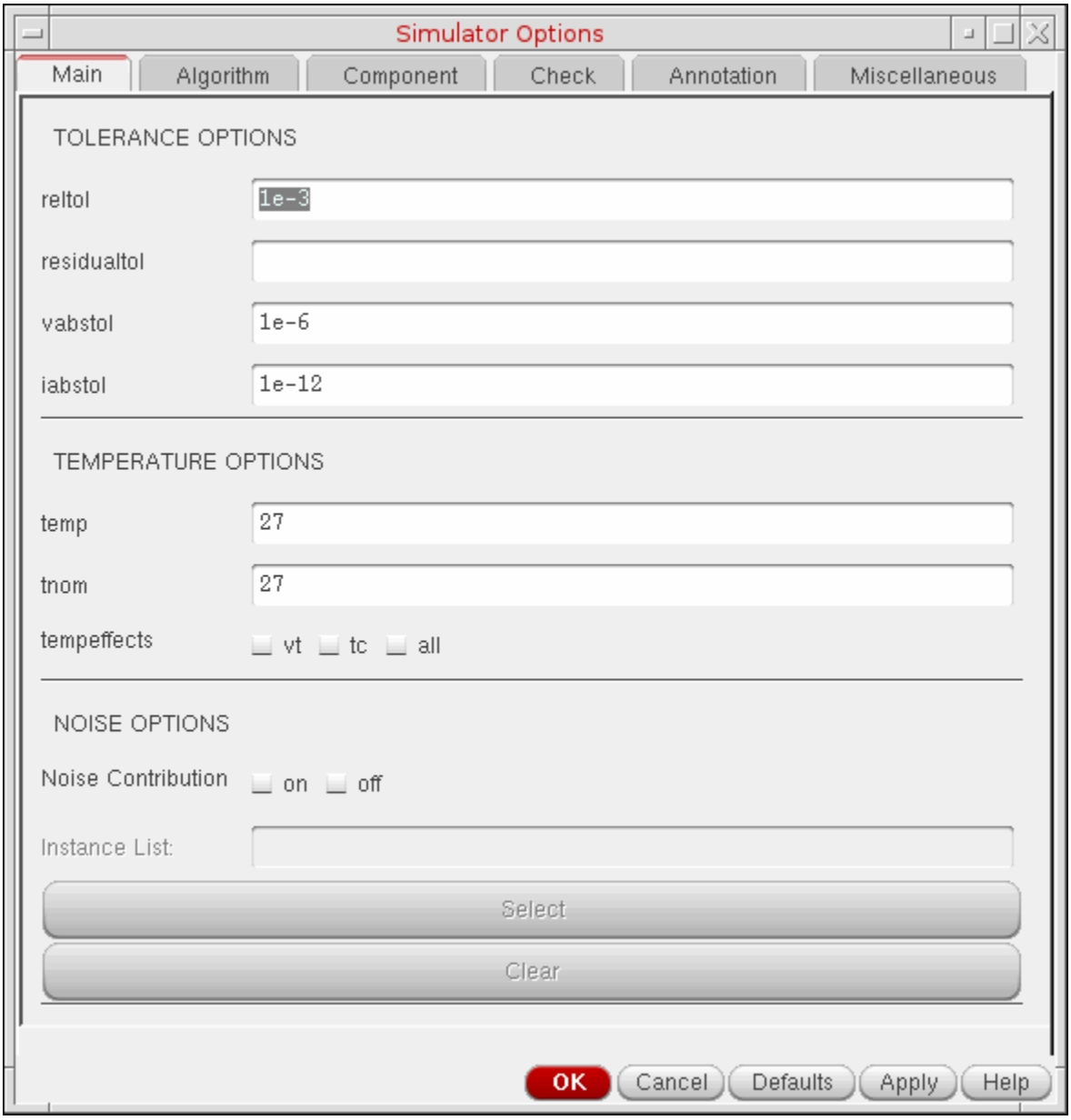

If you are using the Spectre AMS Designer Simulator in IUS 8.1 or later releases, and have selected Spectre as the solver, you can choose *Setup – Performance* to specify performance options.

**Note:** When you change the solver, the values in all GUI fields will be reset to the default values for that solver.


© 2023 Cadence Design Systems, Inc.

### Analog (Spectre)

Choose *Simulation – Options – Analog (Spectre).* The Simulator Options form appears.

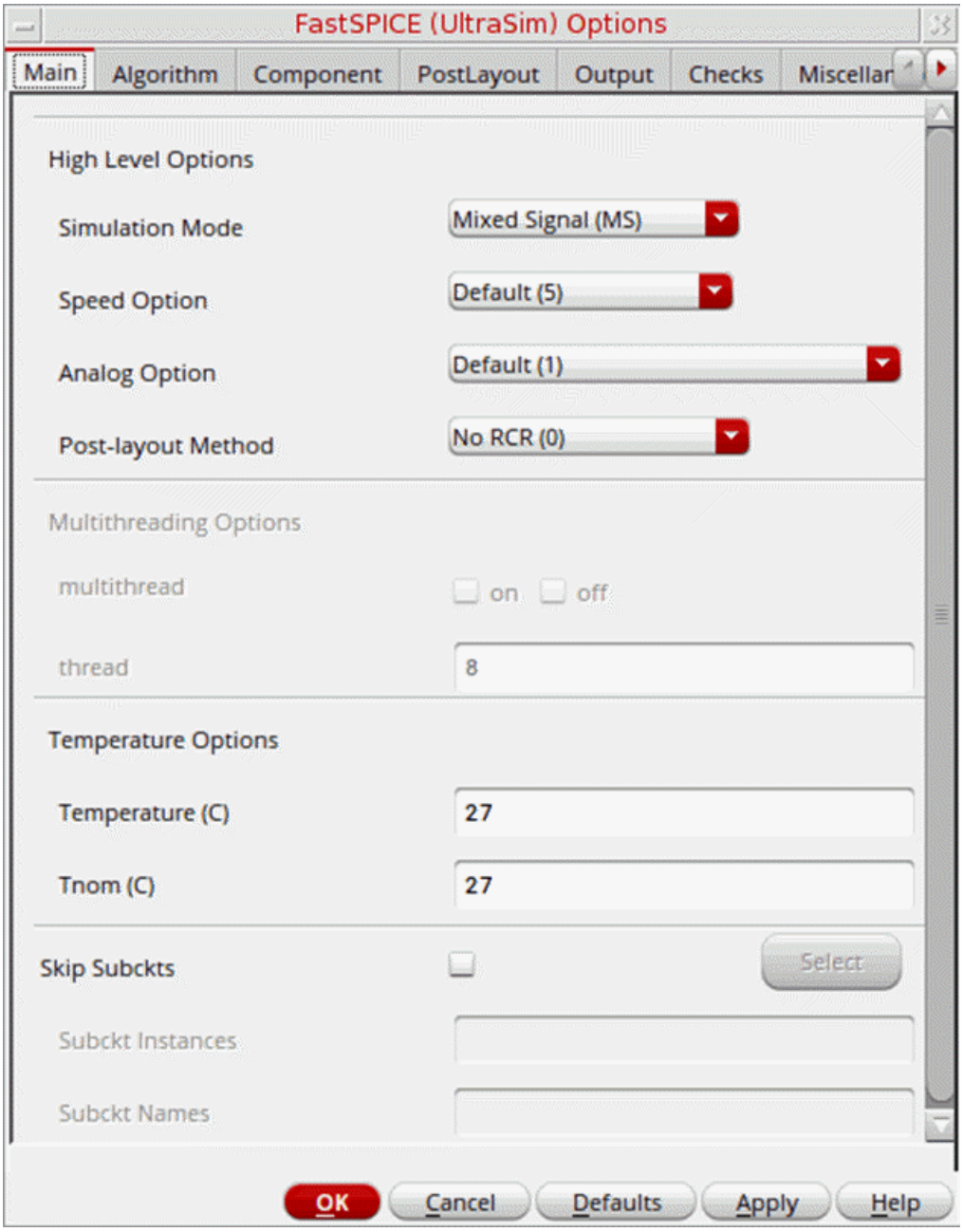



For details, refer to the Immediate Set Options in the *Spectre Circuit Simulator Reference.*


© 2023 Cadence Design Systems, Inc.

### FastSPICE (UltraSim)

Choose *Simulation – Options – FastSPICE (UltraSim).* The FastSPICE (UltraSim) Options form appears.




For more details about the Virtuoso UltraSim simulator options, refer to the *Virtuoso UltraSim Simulator User Guide*.

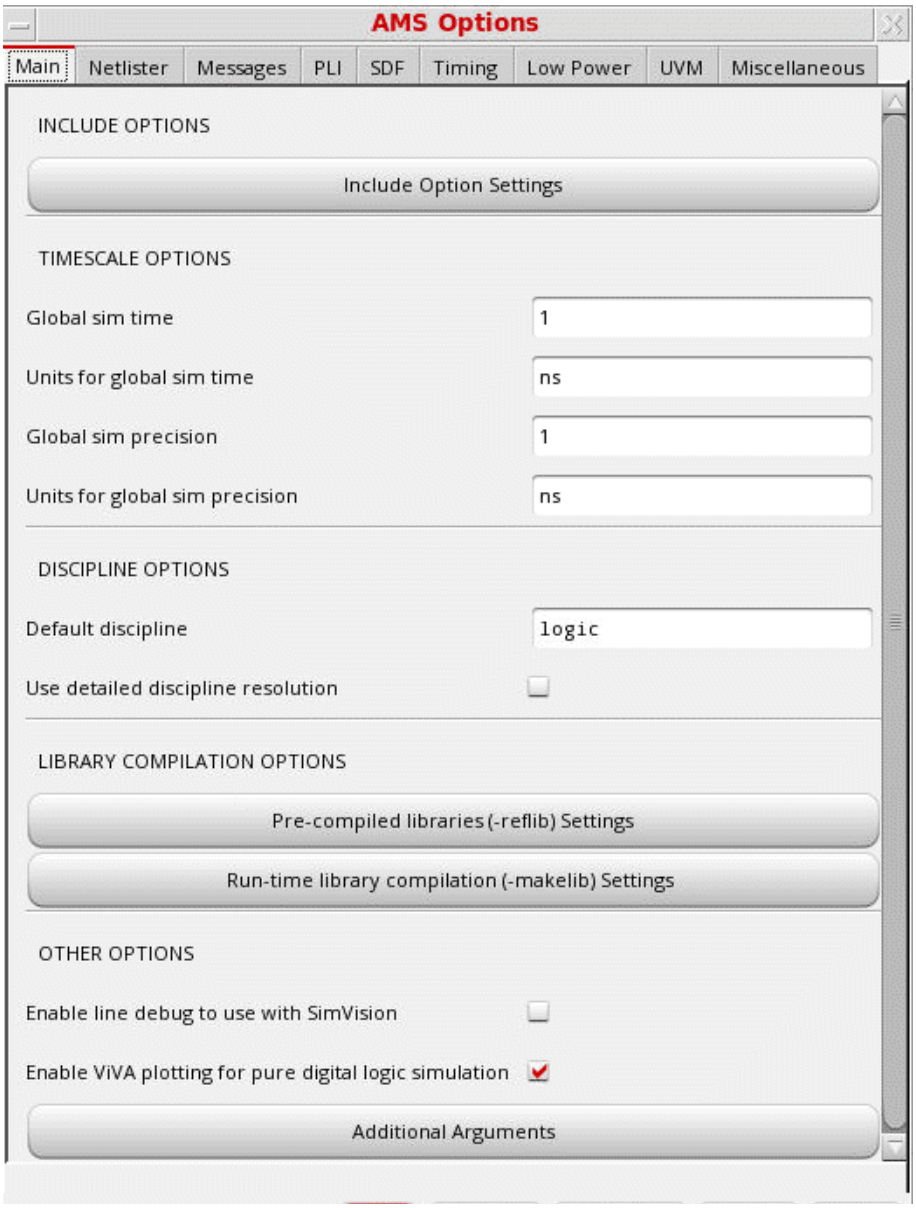

© 2023 Cadence Design Systems, Inc.

## AMS Simulator

You can set the AMS options using the *AMS Options* form. To open this form, choose *Simulation – Options – AMS Simulator*.

The next sections explain the tabs in the *AMS Options* form.

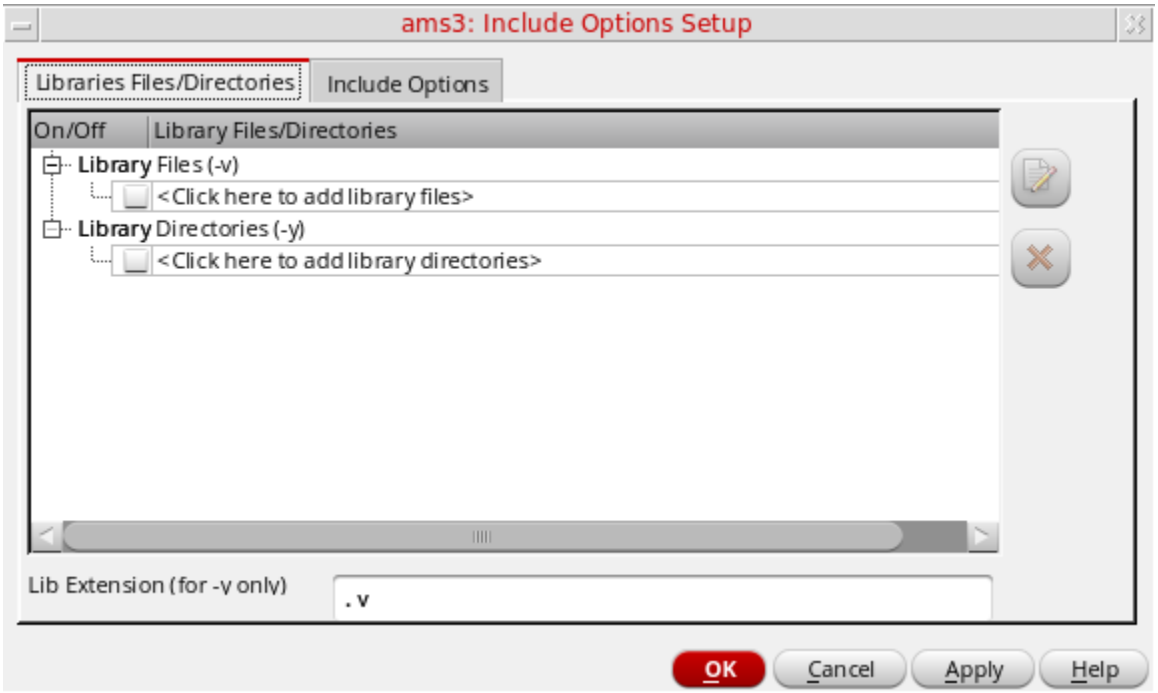
Main tab:




In the *Main* tab, you can specify INCLUDE, TIMESCALE, DISCIPLINE, LIBRARY COMPILATION, and OTHER options.


© 2023 Cadence Design Systems, Inc.

- Click the *Include Option Settings* button. The Include Options Setup form appears.

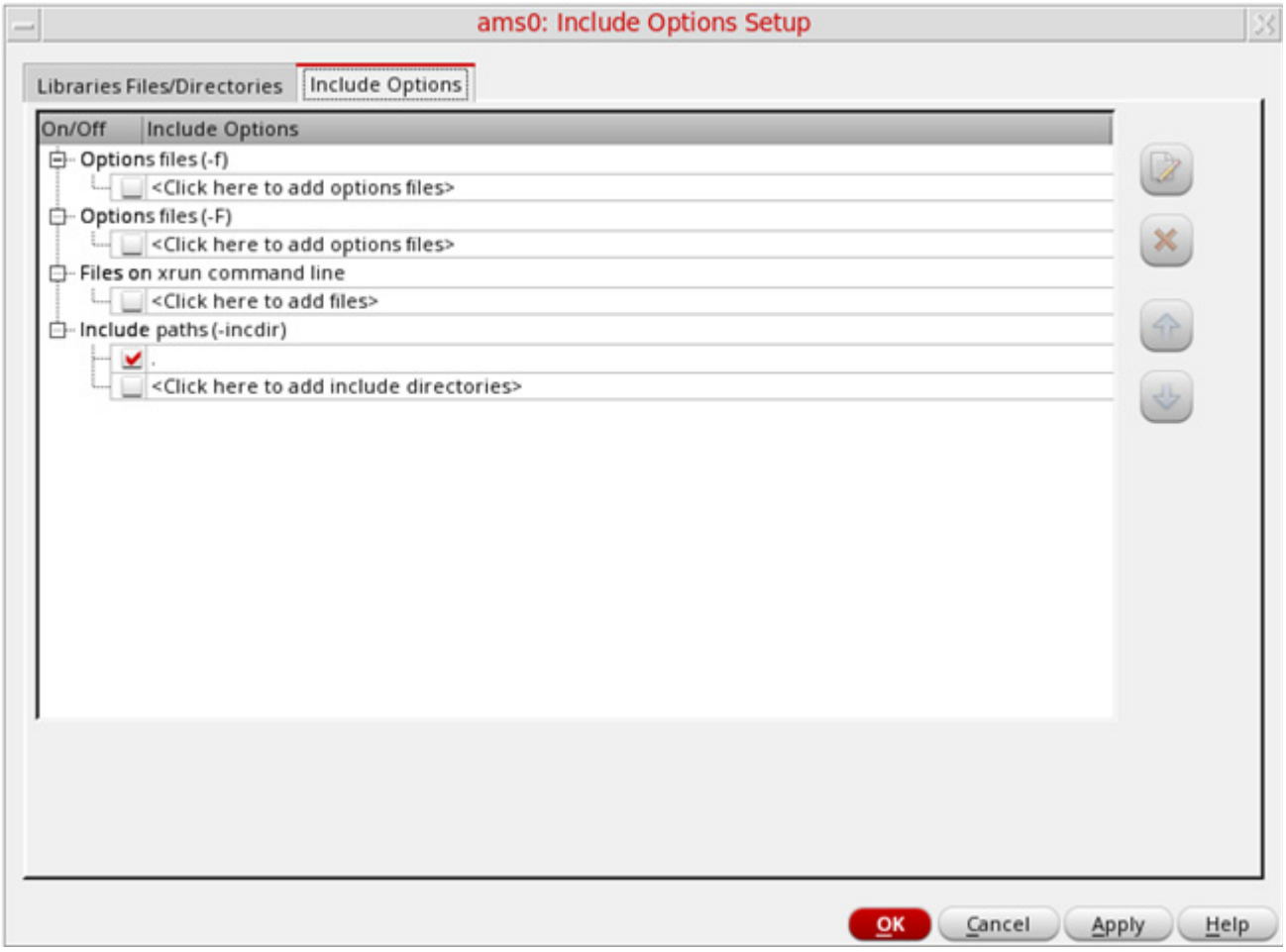



In the *Libraries Files/Directories* tab on the Include Options Setup form, specify the following:

- *Library files (-v)* — Specify the Verilog files to be included for compilation. This field is equivalent to the `-v` command-line option.
- *Library directories (-y)* — Specify the library directories from which the Verilog (.v) files are to be included for compilation. This field is equivalent to the `-y` command-line option. The filenames must match Verilog module names in order to be properly found.
- *Lib Extension (for -y only)* — Specify the extension of Verilog files present in the specified library directories that are to be included for compilation. By default, only `.v` files are included for compilation. Specify the file extensions with a comma-separated list with no space between the entries (`.v,.sv` ). This field is equivalent to the `-libext` command-line option.


© 2023 Cadence Design Systems, Inc.

In the *Include Options* tab on the Include Options Setup form, specify the following:




- *Options file(-f)* — Specify the absolute path of the file from which the command-line options are to be read. This field is equivalent to the `-f` command-line option.
- *Options file(-F)* — Specify the relative path of the file from which the command-line options are to be read. This field is equivalent to the `-F` command-line option.
- *Files on xrun command line* — Specify the files to be added to the xrun command.
- *Include paths (-incdir)* — Specify the directory in which the include files are to be searched. This field is equivalent to the `-incdir` command-line option.

**Note:** You can manually reorder the specified files with the up and down arrows.

- In the TIMESCALE OPTIONS section, specify the following:
  - *Global sim time* — Specify the global simulation time.


© 2023 Cadence Design Systems, Inc.

- *Units for global sim time* — Specify the unit to be used for global simulation time.
- *Global sim precision* — Specify the global simulation precision.
- *Units for global sim precision* — Specify the unit to be used for global simulation precision.

- In the DISCIPLINE OPTIONS section, specify the following:
  - *Default discipline* — Specify the default discipline.
  - *Use detailed discipline resolution* — Specify whether or not you want to use the detailed discipline resolution algorithm.

**Note:** For the *AMS Unified Netlister with xrun* mode, the LIBRARY COMPILATION OPTIONS section is displayed in the *Main* tab.

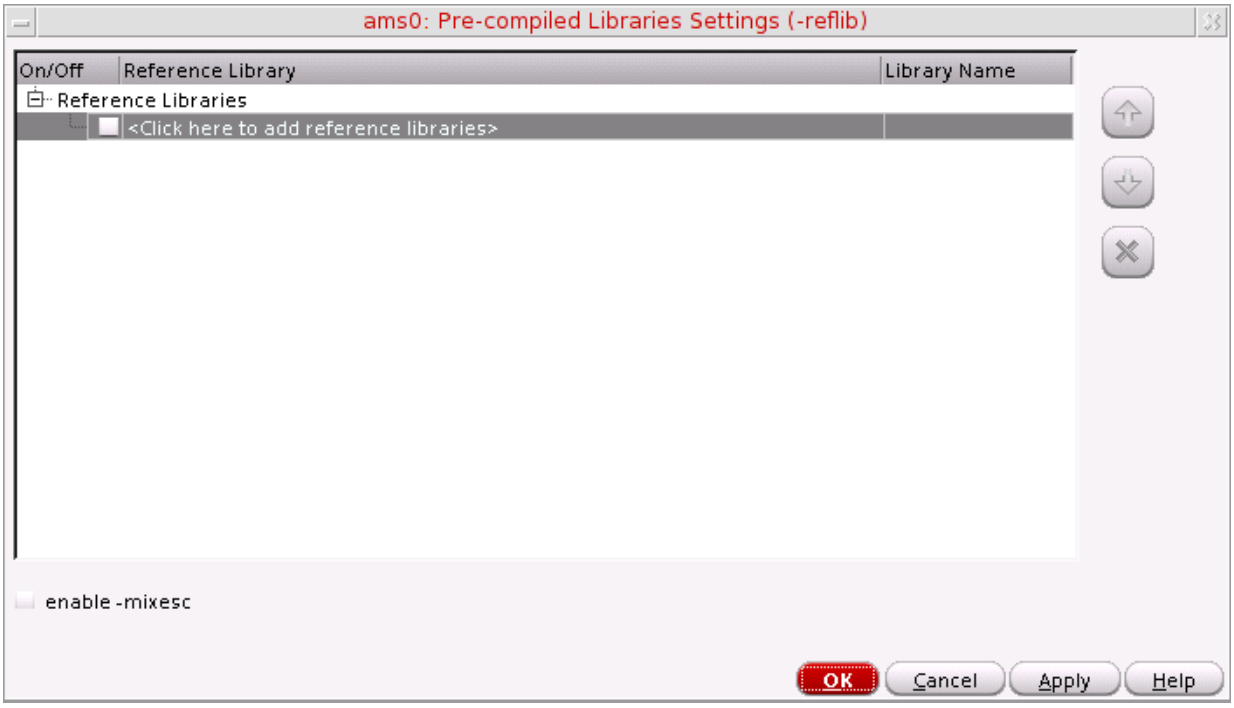
- In the *LIBRARY COMPILATION OPTIONS,* specify the following options:
  - *Pre-compiled libraries (-reflib) Settings:* Specifies the library directory containing precompiled source files that can be referenced during simulation. When you click *Pre-compiled libraries (-reflib) Settings* the *Pre-compiled libraries (-reflib)* form appears where you can add the path to the library directory and specify the logical library name. While specifying reference libraries, you can also use wildcard characters, such as an asterisk (`*`) or a question mark (`?`), to select multiple libraries with similar names.

    **Note:** Ensure that the precompiled libraries that you are using are compatible with the bit version of Virtuoso. A 32-bit Virtuoso release requires 32-bit xrun precompiled libraries, whereas a 64-bit Virtuoso release requires 64-bit xrun precompiled libraries.

    **Note:** Using `-reflib` settings assume that the text libraries were precompiled


© 2023 Cadence Design Systems, Inc.

external to Virtuoso using the `xrun -makelib` command.

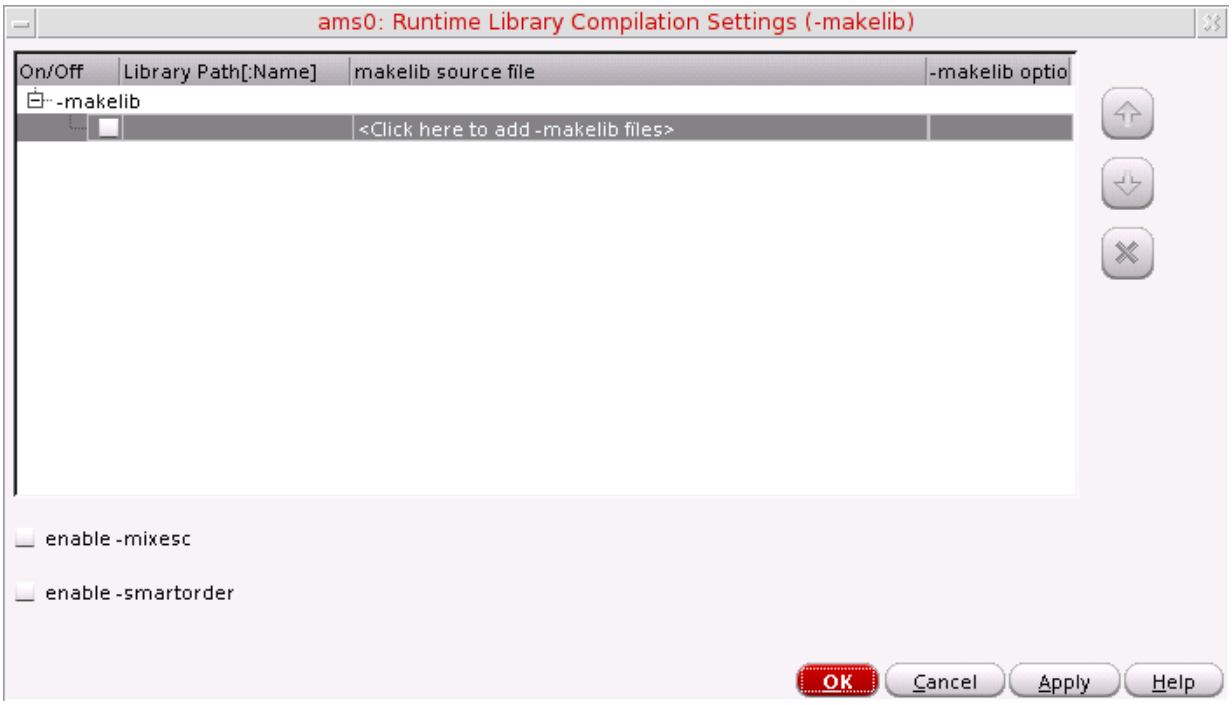



Select the *enable -mixesc* check box to make AMS UNL add the xmelab `-mixesc` option for VHDL text precompiled as mixed-case entity and architecture naming conventions and properly escaped to indicate the special mixed-case syntax. It is important to properly "escape"(\) entity and architecture design unit names within VHDL source code so as to allow proper name mapping in the Virtuoso environment.

- *Run-time library compiling (-makelib) Settings:* Precompiles design units in the specified files into a reference library. When you click *Run-time library compiling (-makelib) Settings*, the *Runtime Library Compilation Settings (-makelib)* form appears, where you can specify the filename(s) containing the design units and the reference library name. While specifying the makelib source files, you can also use wildcard characters, such as an asterisk (`*`) or a question mark (`?`), to select multiple source files with similar names for a single library. Use the makelib options to add library and file specific options needed to compile the design units. If multiple


© 2023 Cadence Design Systems, Inc.

source files have the same library path and makelib options specified in the form, they are added to a single `xrun -makelib` command.




**Note:** Ensure that the makelib options are correctly specified for each makelib source file in the *Runtime Library Compilation Settings (-makelib)* form.

Select the *enable -smartorder* check box to invoke a smart compile order algorithm for order-dependent file compilations.

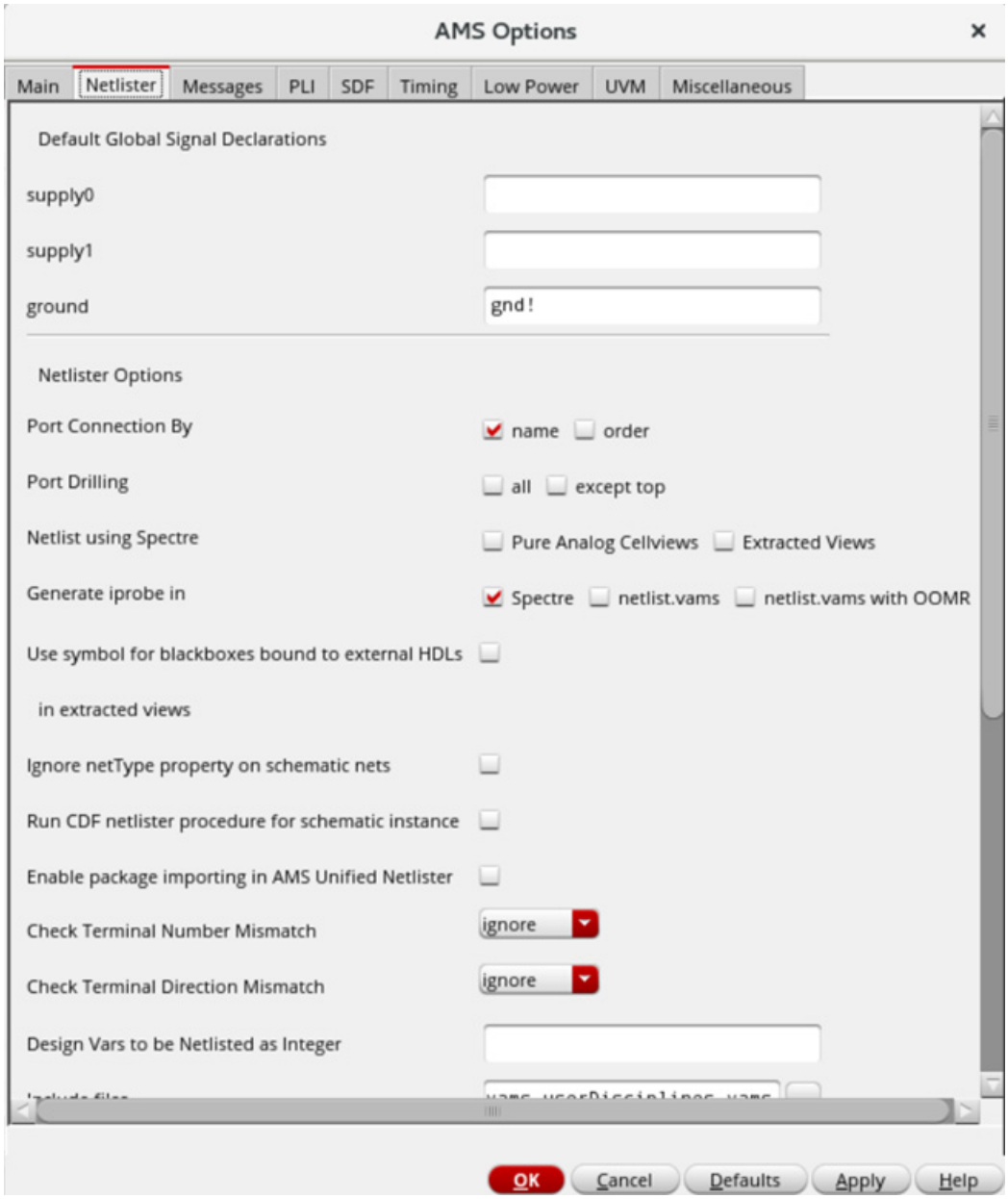
In the *OTHER OPTIONS* section, specify the following:

- *Enable line debug to use with SimVision* — Select this check box to add the `-linedebug` option to the `xrun` command, which enables the support for setting breakpoints at various lines of the code when the *Interactive* simulation mode is selected in the Netlist and Run option form.
- *Enable ViVA plotting for pure digital logic simulation* — Select this check box to add the `-ade` option to the `xrun` command, which enables support for plotting pure digital simulation results with Virtuoso Visualization and Analysis XL. It is enabled as a default setting.
- *Additional Arguments* — Displays the Additional Arguments form that lets you add additional control statements that cannot be specified in the AMS Options form. Arguments entered here appear in the same way as in the generated `xrunArgs`.


© 2023 Cadence Design Systems, Inc.

- If the shell environment variable `DO_NOT_MIGRATE_ADDARG` is set as `YES`, then the string value in *Additional Argument* (before IC6.1.7 ISR17 releases) will no longer be migrated to the *Additional Arguments* form.

Netlister Tab:




In the *Netlister* tab, you can specify DEFAULT GLOBAL SIGNALS DECLARATION and NETLISTER OPTIONS. The following options are available in this tab:


© 2023 Cadence Design Systems, Inc.

- *Default Global Signals Declarations* — Enter a space-separated list of names of global signals in the fields according to how you want to declare them:
  - In the *supply0* field, type names of global signals that you want to declare as type `supply0`.
  - In the *supply1* field, type names of global signals that you want to declare as type `supply1`.
  - In the *ground* field, type names of global signals that you want to declare as type `ground`.
- *Port connection by* — Select the type of port connection.
  - *name* — Select this if you want an explicit connection. Using this option, the terminal names are printed on the primitive or subcircuit terminals and the net name on the corresponding instance terminal in `<termName>=<netName>` format.
  - *order* — Select this if you want an implicit connection. Using this option, the nets connected to instance terminals appear separated by white spaces and in the same order in which master subcircuit or primitive terminals are printed.
- *Port Drilling* — This feature is enabled to generate a hierarchical transistor-level netlist of the design in which inherited connections are converted into explicit power ports throughout the hierarchy. In the scenarios like AMS LP Simulations and current-probing of sub-blocks, and so on, it is beneficial to have explicit power routing through the hierarchy, instead of routing it through inherited connections.

  It has two options: `all` and `except top`.

The following scenarios apply to its flow:

- Cellview containing inherited connections are inside the schematic switch-master

  When you select either `all` or `except top`, and generate a netlist, the inherited connections coming from the schematic would be created as explicit drill ports.

- The top cell is schematic

  When you select `all` and generate a netlist, then the inherited connections coming from the top hierarchy will be created as explicit ports on the `verilogams` module of the top cell. When you select `except top`, then no explicit port will be drilled on the top cell.

- The top cell is a digital cellview

  Explicit ports are not drilled on the digital cellviews.

- The middle cell is text a cellview.


© 2023 Cadence Design Systems, Inc.

Explicit ports are not drilled on the text cellviews.

**Note:** If there are some inherited connections inside a schematic that is below a text cellview in the hierarchy cone, then you should set its net set property on its instance. Virtuoso considers such scenarios as unsupported configuration.

- *Netlist using Spectre*— Specifies whether Spectre can be used to netlist the selected type of cellviews - *Pure Analog Cellviews* or *Extracted View*s in a design, even though the AMS simulator has been set for simulation. Select the either of the following cellview for Analog Only Netlisting.
  - *Pure Analog Cellviews*
  - *Extracted Views*
- *Generate iprobe in*— Specifies where the iprobe statements must be generated.
  - *Spectre*: Generates the iprobe statements in the directory `netlist/analog/ihnl/cds0/iprobeLine.scs` and the iprobe binding information in the directory `netlist/analog/iprobeAmsd.scs`. The binding file is then included in the `spiceModels.scs` file.

    This is the default selection.

    The following examples demonstrate how the iprobe statements are generated:

    - In the `netlist/analog/iprobeAmsd.scs` directory

      ```
      amsd {
      config scope="pl_lib.tb_cell3:schematic" model="analog/ihnl/cds0/
      iprobeLines.scs"}
      ```

    - In `spiceModels.scs` file

      ```
      simulator lang=spectre
      include "analog/iprobeAmsd.scs"
      include "models/spectre/gpdk045.scs" section=mc amsd_subckt_bind=yes
      netlist/analog/ihnl/cds2/iprobeLines.scs
      ```

  - *netlist.vams*: Generates the iprobe statements in the `netlist.vams` file.
  - *netlist.vams with OOMR*: Generates the iprobe statements in the `netlist.vams` file and sets the shell variable `ASSEMBLER_SUPPORT_OOMR` to `YES`.
- *Use symbols for blackboxes bound to external HDL*—Select this check box to allow the netlist to use a symbol view to generate an instance line for a blackbox in an extracted view. This blackbox must have the external HDL properties set in the *HED* (*Hierarchy Editor*).

  Related environment variable: netlistBlackboxWithExtHdl


© 2023 Cadence Design Systems, Inc.

- *Run CDF netlister procedure for schematic instance* — Select this check box to enable the Unified Netlister to run a netlister procedure on a schematic. For this ensure that a netlister procedure is specified in spectre simInfo in the CDF parameters of the selected cell.
- *Create Spectre subcircuit for extracted view* — Select this check box to enable Unified Netlister to generate the Spectre subckt definition for the extracted view into the `pureAnalog.scs` file passed to the simulator with the `netlist.vams` file.

  **Note:** This feature is available only for analog netlisting on the extracted view.

  The following scenarios apply to this flow:

  - If the extracted view is a pure analog cell, the Spectre subckt definitions are printed in the `pureAnalog.scs` file.
  - If the extracted view contains an instance bound to a pure analog hierarchical view, the Spectre subckt definitions for all the hierarchical cells in the partial hierarchy are printed in the `pureAnalog.scs` file.
  - If the extracted view contains an instance bound to a digital view, the netlister prints an error message and does not create the netlist.
- *Ignore netType property on schematic nets* — Select this check box to stop printing the `netType` property from the design schematic to the `netlist.vams` file.
- *Enable package importing in AMS unified Netlister* — Select this check box to enable the auto-package handling feature. It allows Unified Netlister to automatically search and find the `systemverilog/vhdl/vhdlams` package files on which the `systemverilog/vhdl/vhdlams` modules in the design depend.
- *Check Terminal Number Mismatch* — Specify the type of message to be generated when the number of terminals on the module or cellview and the number of terminals on the instance do not match. It can be set to the following:

  - *ignore* — If selected, all error messages and warnings are ignored. This is selected by default.
  - *warning* — If selected, generates a warning message. The netlister does not stop netlisting if a warning message is generated.
  - *error* — If selected, generates an error message. The netlister stops netlisting when an error message is generated.

  **Note:** For instances with a master text cellview, UNL netlists only those terminals that are needed according to the master cellview. And the type of the message to be generated is determined by the value of the `simCheckTermMismatchAction` variable specified in the `.simrc`.


© 2023 Cadence Design Systems, Inc.

- *Check Terminal Direction Mismatch* — Specify the type of message to be generated when the direction of cellview ports and the connection on the instance do not match. It can be set to the following:
  - *ignore*—If selected, all error messages and warnings are ignored. This is selected by default.
  - *warning*—If selected, generates a warning message. The netlister does not stop netlisting if a warning message is generated.
  - *error*—If selected, generates an error message. The netlister stops netlisting when an error message is generated.
- *Design Vars to be netlisted as integer* — Specify the list of design variables that should be processed as integers.
- *Include files* — This field allows you to include the contents of the specified files that are required for the design to simulate. Click the ellipses button to select the files.
- *Maximum number of warnings* — Set the maximum number of warnings issued by the netlister before it stops processing the design.

  **Note:** You can also specify the maximum number of warnings using the *netlistMaxWarn* environment variable.
- *Suppress specific netlister info/warnings* — Specify a space-separated list of message IDs, such as `AMS-2000 AMS-2171 AMS-2174`, that you want to suppress while processing the design.

  **Note:** You can also suppress the information or warning messages by using the *netlistNoWarn* environment variable.
- *Upgrade specific netlister messages to WARNING* — Specify a space-separated list of information message IDs, such as `AMS-1244 AMS-1246`, that you want to be treated as warning messages while processing the design.

  **Note:** You can also upgrade the information messages to warning messages by using the *upgradeMsgSevWarn* environment variable.
- *Upgrade specific netlister messages to ERROR* — Specify a space-separated list of information and warning message id, such as `AMS-2171 AMS-2174`, that you want to be treated as error messages during the processing of the design.

  **Note:** You can also upgrade the warning messages to error messages by using the *upgradeMsgSevError* environment variable.
- *Netlist using spectre CDF simInfo* — Specify view names that will be treated as analog cells during netlisting. The specified view names are netlisted using the cell's


© 2023 Cadence Design Systems, Inc.

spectre CDF simInfo and are processed as analog stopping view or analog leaf-level primitive.

- *Header text* — This field allows you to specify header text that you want the AMS netlister to insert in every Verilog-AMS.
  - *file* — Select this option to include header text from a particular file.
  - *script* — Select this option to include header text that results from a script file.
- *Global Signals* — This button brings up the Global Signals form, which is described in the next topic.

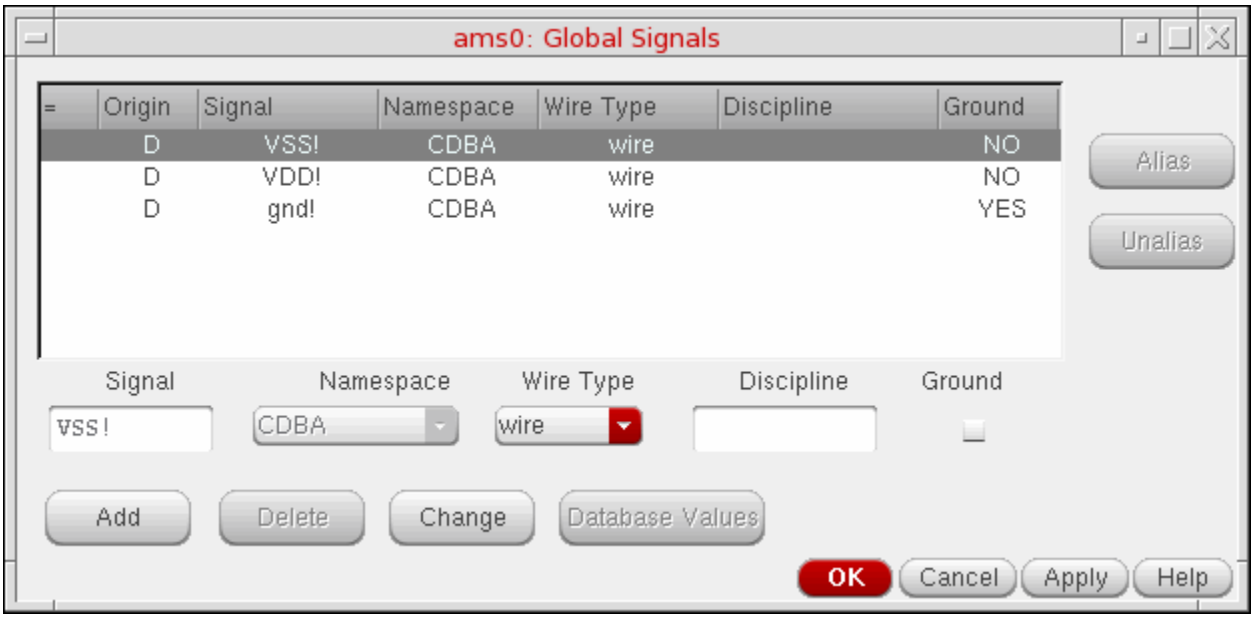
## Working with Global Signals in AMS

A global signal is a signal that is connected by name across all levels of a design hierarchy without using pins. Global signals can come from schematic data or from text modules. AMS is aware of only global signals that come from schematic data. You can use the Global Signals form to declare a signal that is used as an out-of-module signal reference.

To add a global signal,

1. In the AMS Options form, on the Netlister tab, click the *Global Signals* button.

   The Global Signals form appears. If the design has not been netlisted after recent changes, you are prompted to netlist the design so that the Global Signals form can display the latest data.





© 2023 Cadence Design Systems, Inc.

2. Global signals are displayed in a tabular format. It includes the following information.
   - The first column indicates the alias of aliased signals. They represent:
     - The beginning of the aliased set: `/--`
     - The aliased signals in between: `|--`
     - The end of the aliased set: `\--`
   - The *Origin* column is either blank if the signal was added using the Global Signals form, or it has the value `D` to indicate that a signal has been extracted from the design.
   - The *Signal* column shows the name of the signal.
   - The *Namespace* column displays the namespace in which the signal was created.
   - The *Wire Type* column shows the wire type of the signal.
   - The *Discipline* column shows the discipline of the signal.
   - The *Ground* column indicates if the global signal is used as a ground reference.

   **Note:** Netlisting extracts the signals in the design and merges them with the signals created using the Global Signals form. If you open the form without netlisting, it would be empty.
3. The input fields below the report get populated by signal details when you select any signal. To change these values, you can either type over them or select values from the cyclic lists and then click the *Change* button.
4. To add new global signals,
   a. Type a unique name for the signal in the *Signal* field. You can specify a range such as `<5:8>` by post-fixing it to the name.
   b. Select the namespace as *CDBA*, *Spectre*, *Spice* or *Verilog-AMS* from the *Namespace* cyclic list.
   c. Select one of these from the *Wire Type* cyclic list: *wire, supply0, supply1, tri, tri0, tri1, triand, trior, trireg, wand, wor,* or *wreal.*

      **Note:** If the signal name you specify is included in the *supply0* or *supply1* field of the Netlister Options form, the default value for wire type would be changed to *supply0* and *supply1* accordingly.
   d. Type a discipline name for the signal in the *Discipline* field.


© 2023 Cadence Design Systems, Inc.

**e.** If you want to use the global signal as a ground reference, select the *Ground* option. You can select this option only for signals that have the wire type *wire* or *tri*.

**Note:** If the signal name you specify is included in the *ground* field of the Netlister Options form, this field appears selected by default.

**f.** Click the *Add* button.

The new global signal appears in the list of global signals.

**Note:** You can create a new global signal from an existing one by selecting one, modifying its values and clicking *Add*.

**5.** To delete one or more signals, select them and click the *Delete* button.

You cannot delete a global signal that is extracted from the design. If you select such a signal, the Namespace and Name fields appear non-editable. If you change any of the other values, the *Database Values* button is enabled, using which you can set the fields back to their original values from the database.

**6.** You can alias global signals into groups. Aliased signals in a group are electrically equivalent, as if they are joined by a wire. To alias global signals, select the signals to be aliased and click the *Alias* button.

To select signals listed consecutively, hold down the `Shift` key while you click the signal names to be aliased. To select signals that are not listed sequentially, hold down the `Ctrl` key while you click the signal names.

When you alias signals, they redisplay consecutively in the global signal list, joined by a vertical connecting bar. If you alias signals belonging to separate aliased signal groups, all of the signals in the groups are aliased.

**7.** To unalias signals, select the signals to be unaliased from the group, and click the *Unalias* button. If an alias set has only two signals, and you unalias one, the other also gets automatically unaliased.

**8.** When you have finished editing the list of global signals, click *OK*.

You need to regenerate the netlist so that the changes made in this form reflect in the netlist and cds_globals module is regenerated. If you try to create or re-create the netlist without applying the changes in the Global Signals form, your changes get overwritten by the netlist. This is the reason a prompt appears as follows:

```
While netlisting, the globals would again be extracted from the design and the
globals form would be updated. Unsaved changes, if any on the globals form would
be lost. Proceed with netlisting?
```

Netlisting includes the following actions:


© 2023 Cadence Design Systems, Inc.

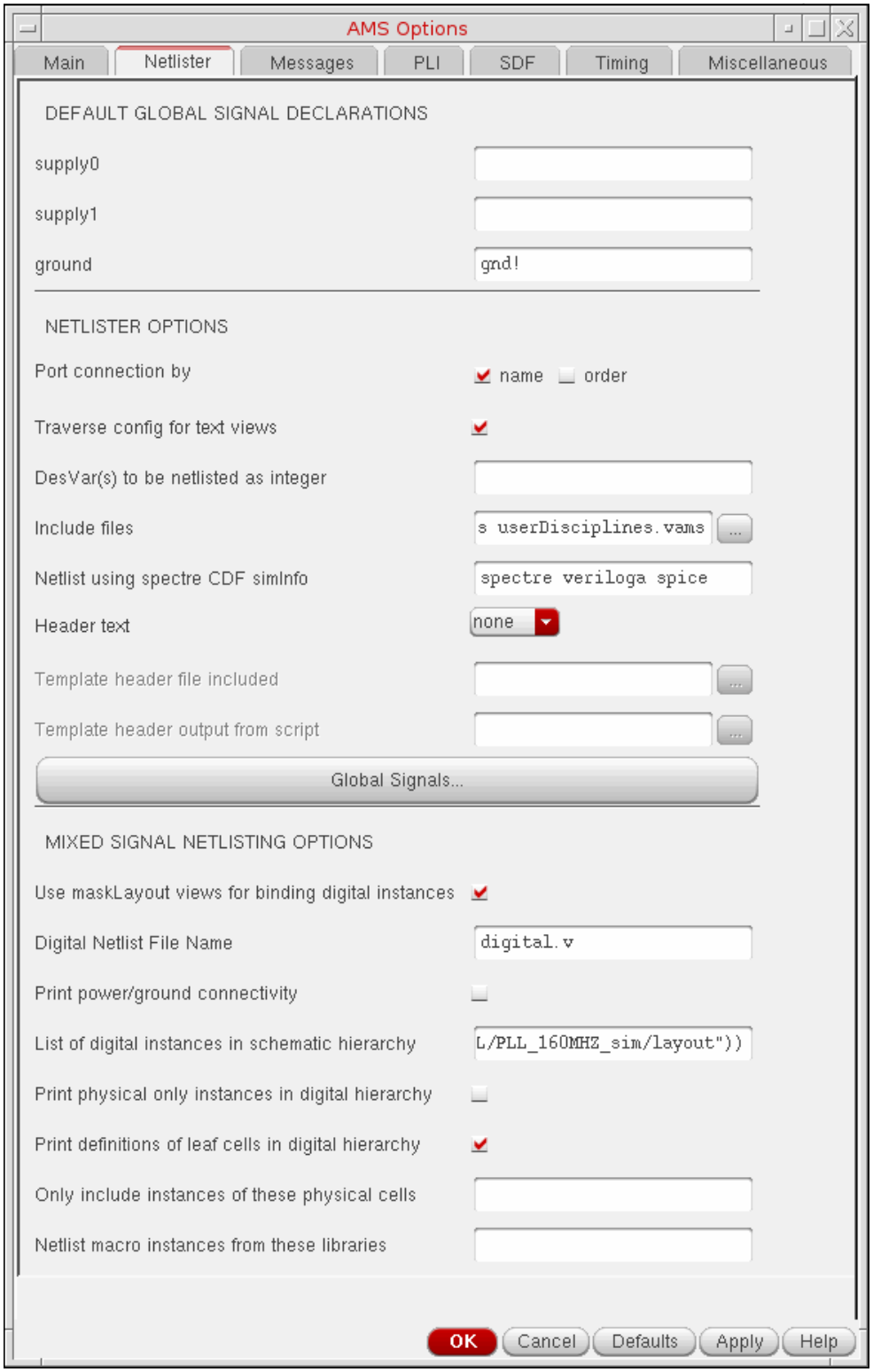
1. It extracts information about all the global signals and design variables from the design.
2. Information about variables is merged with the design and written as it is to the `verilog.vams` file.
3. The global signals information is updated with any new extracted global signals that you modified in the current session.
4. If an extracted global signals exists in the global signals information and its origin is marked as D, and it has not been modified in the design, it is not copied.
5. If an extracted global signals exists in the global signals information and its origin is marked as D, and you modified it using the Global Signals form in the current session, the values are copied over and a message appears saying so.
6. If it exists in the global signals information for the session and the Origin is not marked as ‘D’, a prompt appears as follows:

   ```
   A global with the name '%s' already exists in the session and is now also found
   in the design. Using the one that exists in the session.
   ```

The Global Signals form appears updated the next time it is opened.

**Note:** The *Mixed Signal Netlisting Options* are available in the *Netlister* tab of the *AMS Options* form for AMS simulator only when the *AMS Unified Netlister with xrun* option is selected from the *Netlist and Run Options* form.


© 2023 Cadence Design Systems, Inc.





© 2023 Cadence Design Systems, Inc.

These options enable the Embedded Module Hierarchy (EMH) CDL netlister to generate the digital and analog CDL netlists, simultaneously.

The *Use maskLayout views for binding digital instances* field allows you to enable the EMH netlisting.

The *Digital Netlist File Name* field allows you to specify the name of the output file.

The *Print power/ground connectivity* field allows you to print the power or ground connectivity across the digital module hierarchy. This does not impact the top-level netlist.

The *List of digital instance in schematic hierarchy* field allows you to specify the list of digital instances in the design.

This field is automatically populated if the top-level design contains instances with the `lxEMHStart` property. You will need to update the mapped name to ensure the mapping is correct. For example, `("L/C/Sch" "I0" "L/C/Lay")` should be replaced with `("L/C/Sch" "I0" "L/C/Lay" "|I0")`.

If the top-level design does not contain any instance having the `lxEMHStart` Boolean property, the *List of digital instances in schematic hierarchy* text box is automatically populated without the instance names. For example, if `("DemoLib/top/schematic" "" "DemoLib/top/layout")`, you need to update the instance names in this text box.

It is recommeded that before generating a netlist, you review the list to verify that it contains only valid SKILL forms and update accordingly. You can choose to generate the netlist in any one of the following way:

- Generate the netlist for the entire EMH contained in the specified layout view.
- Generate the netlist for the module hierarchy corresponding to the digital instances listed in this text box.

Sample inputs:

- `I0` and `I1` are two digital instances in the schematic hierarchy

```
(("DemoLib/top/schematic" "I1" "DemoLib/block/layout") ("DemoLib/
mid/schematic" "I0" "DemoLib/top/layout"))
```

- `I0` is a digital instance in schematic, but has a mapped name `I0` in the target layout cellview. This means that `I0` is bound to the master module of the instance `I0` in the layout module hierarchy.

```
(("DemoLib/top/schematic" "I0" "DemoLib/block/layout"  "|I0"))
```


© 2023 Cadence Design Systems, Inc.

- `I0` is a digital instance in top-level schematic, and is bound to the top-level module in the layout-embedded module hierarchy.

```
(("DemoLib/block/schematic" "I0" "DemoLib/block/layout"  t))
```

The *Print physical only instances in digital hierarchy* field allows you to print the physical only instances in digital module hierarchy.

The *Print definition of leaf cells in digital hierarchy* field prints the empty standard cells in the digital netlist.

The digital module hierarchy in the layout contains instances of standard cells. By default, these instances are considered as stopping instances, and the mixed-signal netlister does not print the subcircuit definition for standard cells. If you have the standard cell definitions available from the foundry, you can directly include these definitions during LVS or SVS.

However, if the *Print definition of leaf cells in digital hierarchy* option is selected, the mixed-signal netlister prints the empty standard cell definitions in the digital netlist.

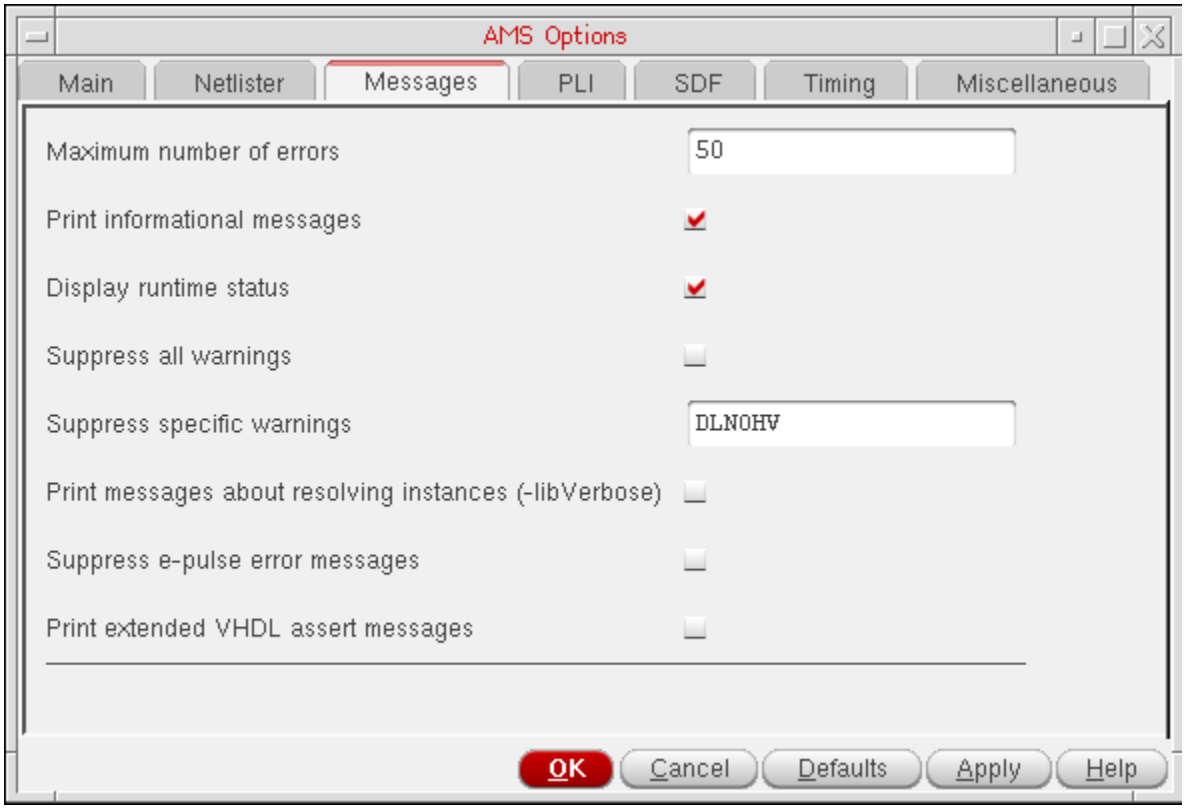
The *Only include instances of these physical cells* field allows you to specify the physical instances that need to be printed in the netlist. If you do not specify a cell name, then all the physical instances are printed in the netlist.

You can also use `*` wildcard as a suffix when specifying cell names, such as `FILL*`, `MFILL*`. The physical instances of cells matching the specified pattern are printed in the netlist. However, other physical-only instances are ignored.

The *Netlist macro instances from these libraries* field allows you to specify the libraries from the `cds.lib` file. The instances of analog schematic cells from the specified libraries are considered hierarchial instances and are netlisted hierarchically by the mixed-signal netlister.


© 2023 Cadence Design Systems, Inc.

## Messages Tab




Using the *Messages* tab, you can control the messages for xmvlog, xmelab, xmsim and xrun. In this tab, you can specify the following options:

*Maximum number of errors*: Number of error messages for xmvlog, xmelab, and xmsim.

*Print informational messages:* If selected, displays the message from the tool.

*Display runtime status:* If selected, displays the runtime status.

*Suppress all warnings:* Suppresses all warnings. If this field is selected, the *Suppress all* warnings field is disabled.

*Suppress specific warnings:* Suppresses warnings with a code.

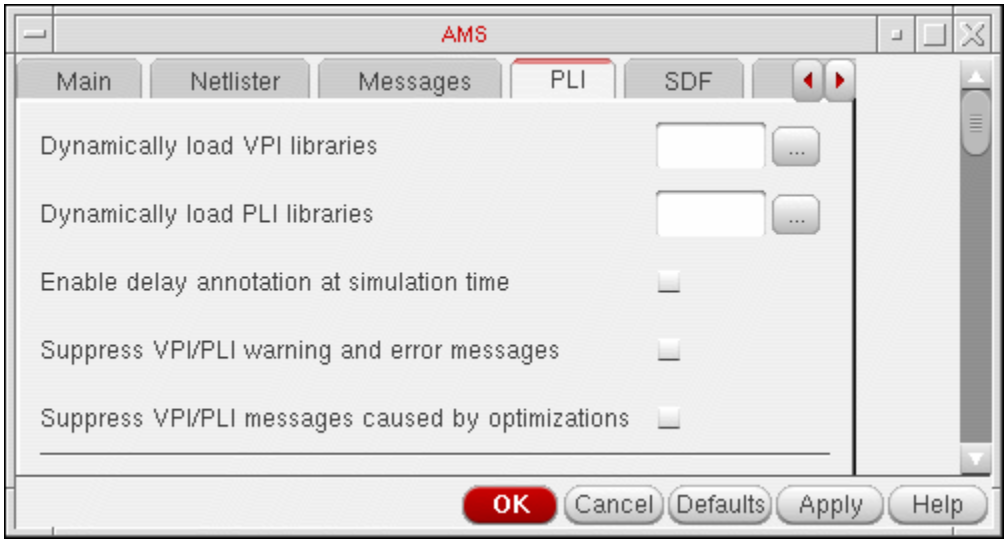
*Print Messages about resolving instances:* This option prints informative messages during execution.

*Suppress e-pulse error messages:* This option suppresses pulse control error messages.


© 2023 Cadence Design Systems, Inc.

*Print extended VHDL assert messages:* This option displays VHDL assert messages with additional information specifying the location in code from where the function or procedure is being called.

Select the *PLI* tab.




In the *PLI* tab, you can specify the following options:

*Dynamically load VPI libraries:* Dynamically loads the specified VPI application.

*Dynamically load PLI libraries:* Dynamically loads the specified PLI application.

*Enable delay annotation at simulation time:* This option disables the optimization in the simulator that take delays into account. Use this option if you intend to modify delays at simulation time. Using this option sets the default access to simulation objects to read/write when the design is elaborated.

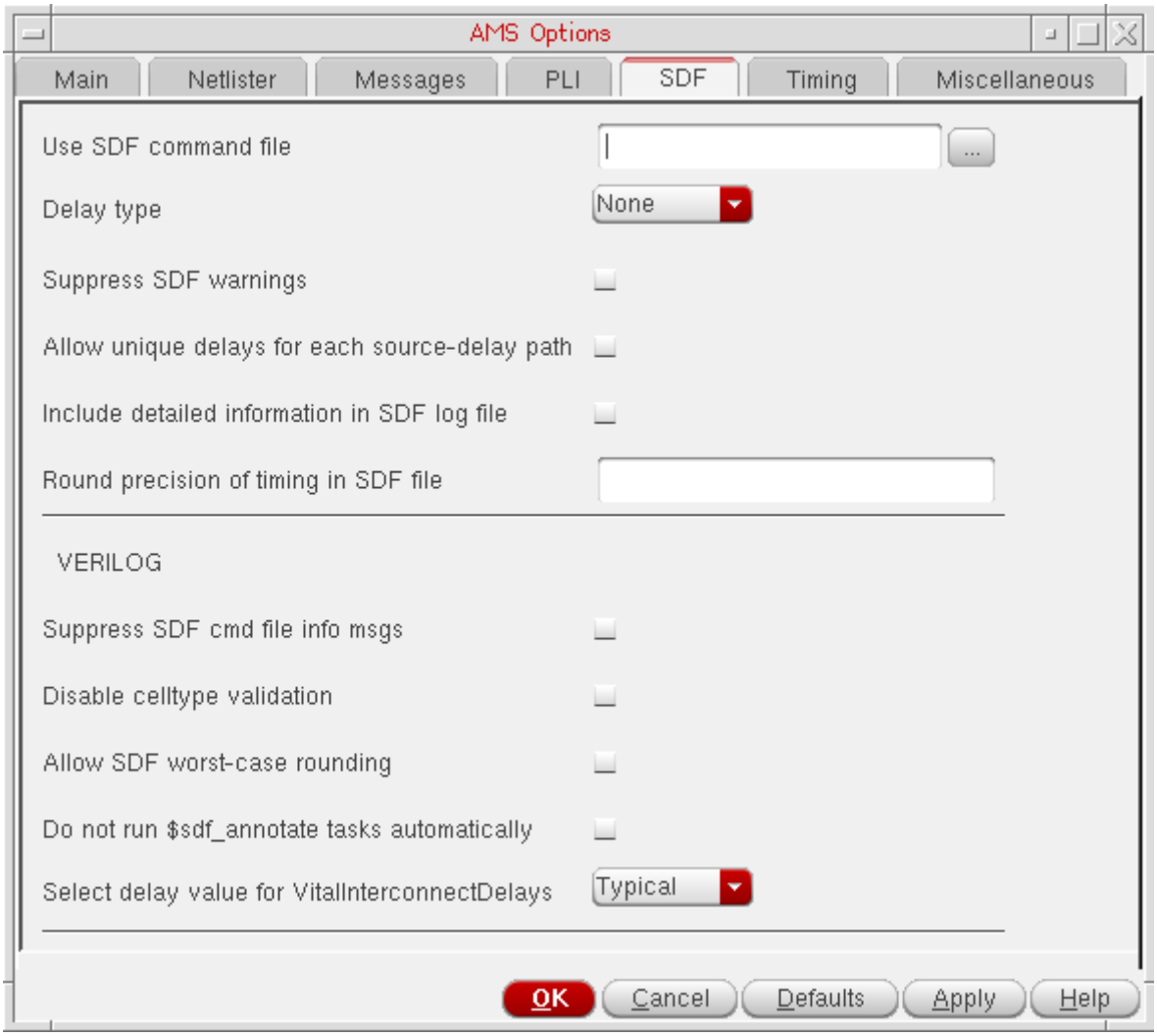
*Suppress VPI/PLI warning and error messages:* Disables printing of PLI warning and error messages.

*Suppress VPI/PLI messages caused by optimizations:* Prints a warning message only the first time that a PLI read, write, or connectivity access violation is detected.

*Disable constraint checking in VDA applications:* The VDA library is a C interface library that provides a mechanism for interaction between other C interface libraries and the objects, scopes, and data types in the VHDL model. VDA also contains routines for examining and manipulating the values of VHDL objects, as well as for setting up callbacks on signal events and simulation times. See *Chapter 6, The VHDL Design Access Library*, in the *Cadence NC-VHDL Simulator C Interface* User Guide for details on VDA functions.


© 2023 Cadence Design Systems, Inc.

### SDF Tab




In the *SDF* tab, you can specify the following options:

*Use SDF command file:* Specifies the command file to be used.

*Delay type:* Enables you to select the delay type. By default, it is set to None.

*Suppress SDF warnings:* Disables the SDF warnings.

*Allow unique delays for each source-delay path:* Enables or disables unique delays for each source-delay path.

*Include detailed information in SDF log file:* Allows you to print information in SDF log file.


© 2023 Cadence Design Systems, Inc.

*Round precision of timing in SDF file:* Allows for precision of timing in SDF file.

In VERILOG section, you can specify the following options:

*Suppress SDF cmd file info msgs (Verilog):* Allows printing of command file information messages.

*Disable celltype validation (Verilog)*: Enable or disable celltype validation.

*Allow SDF worst-case rounding (Verilog):* Allows for worst-case rounding of SDF.

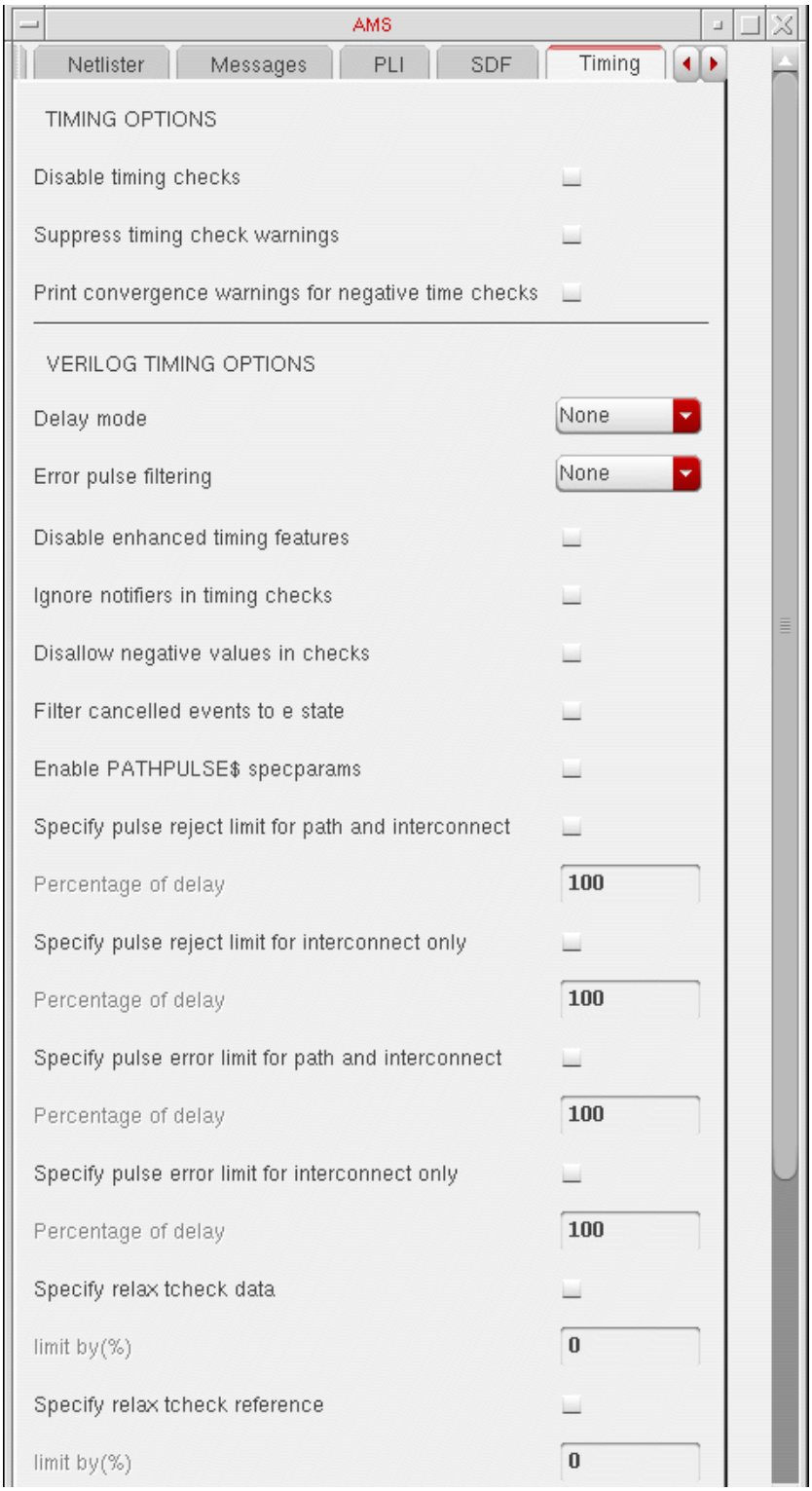
*Do not run $sdf_annotate tasks automatically (Verilog):* Allows you to disable the automatic sdf_annotation tasks.

In VHDL section, you can specify the following option:

*Select delay value for VitalInterconnectDelays (VHDL):* Specifies minimum or maximum delay value to be used during VITAL SDF annotation.

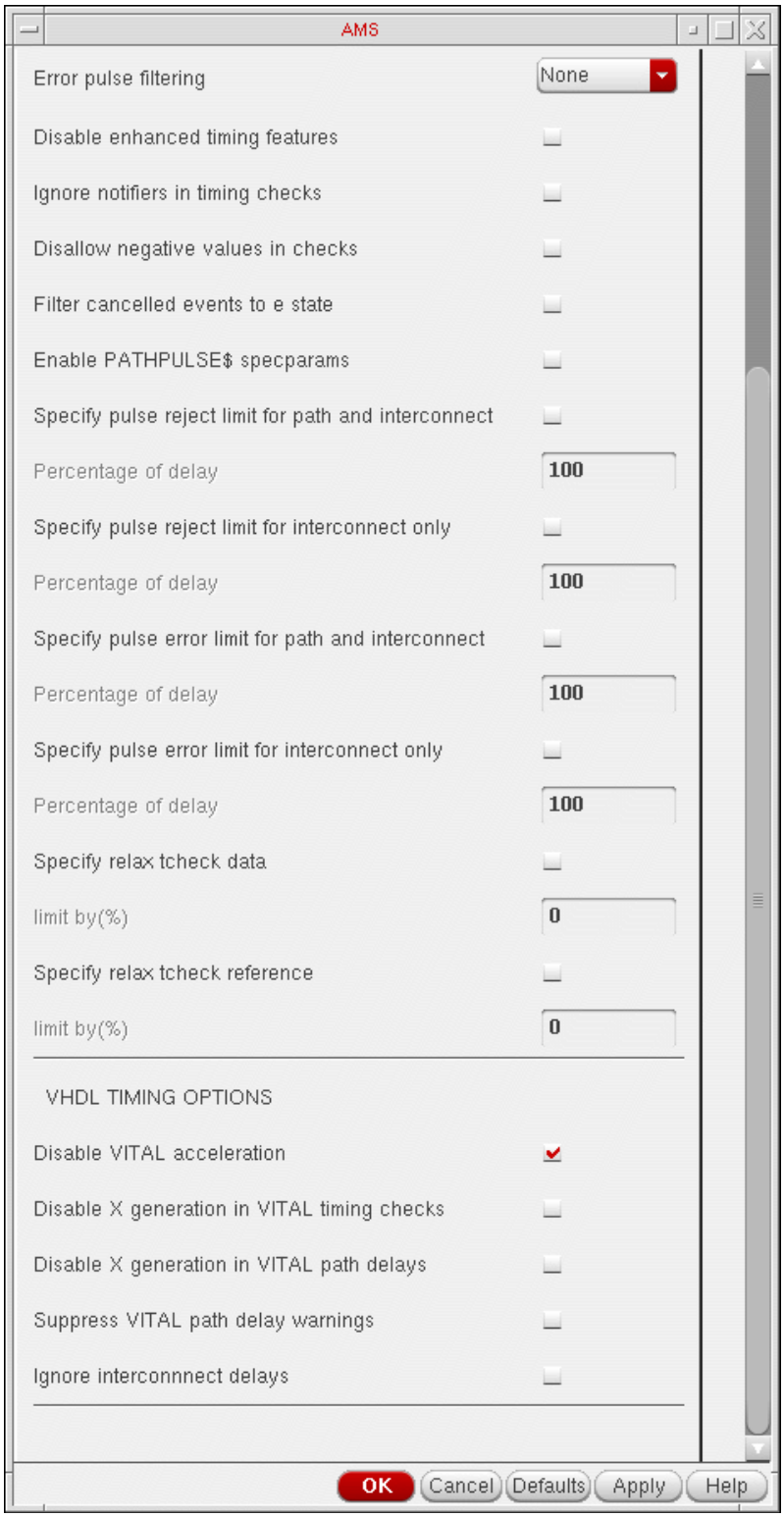

© 2023 Cadence Design Systems, Inc.

## Timing Tab





© 2023 Cadence Design Systems, Inc.



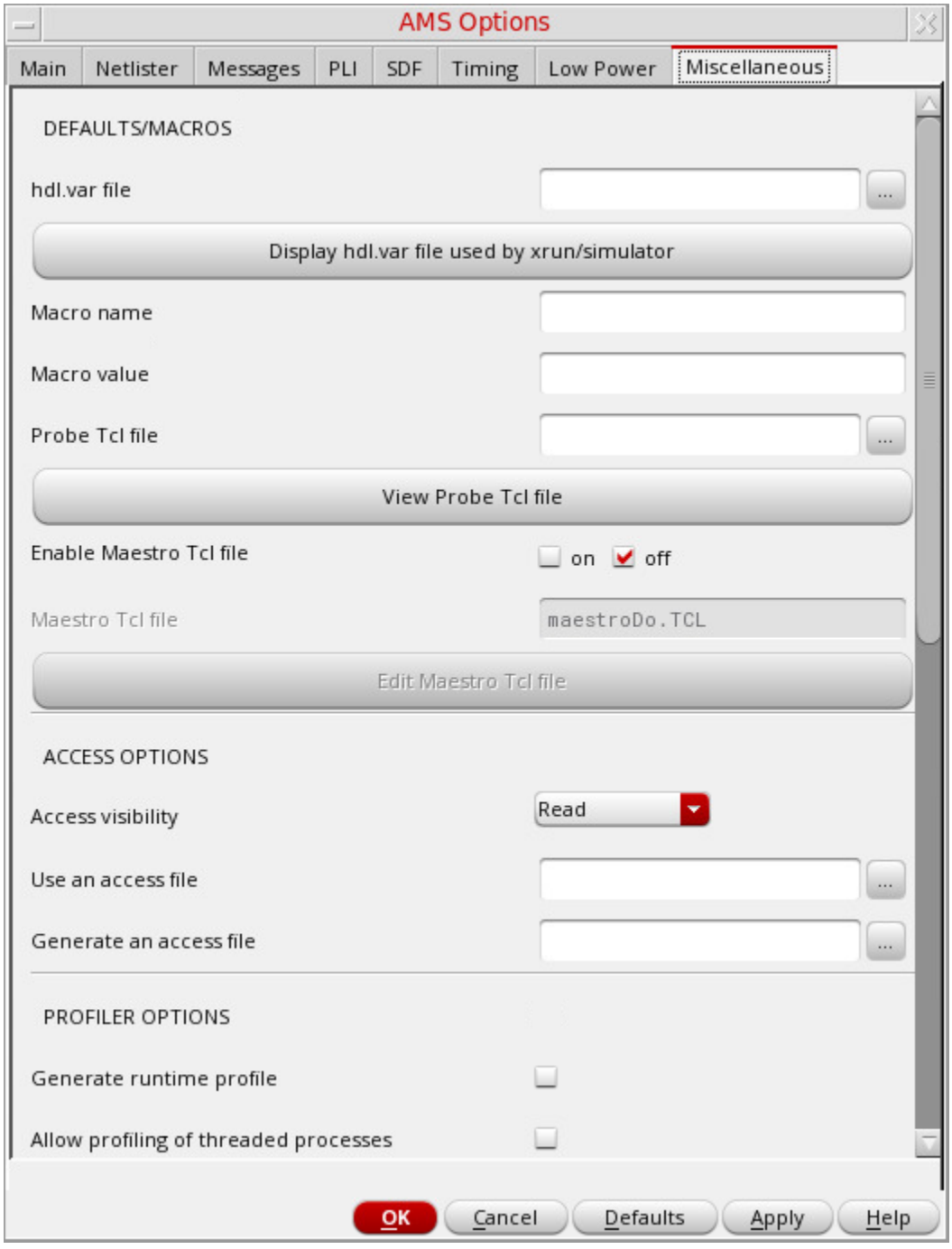


© 2023 Cadence Design Systems, Inc.

In the *Timing* tab, you can specify the timing options for VERILOG and VHDL.


© 2023 Cadence Design Systems, Inc.

## Miscellaneous Tab

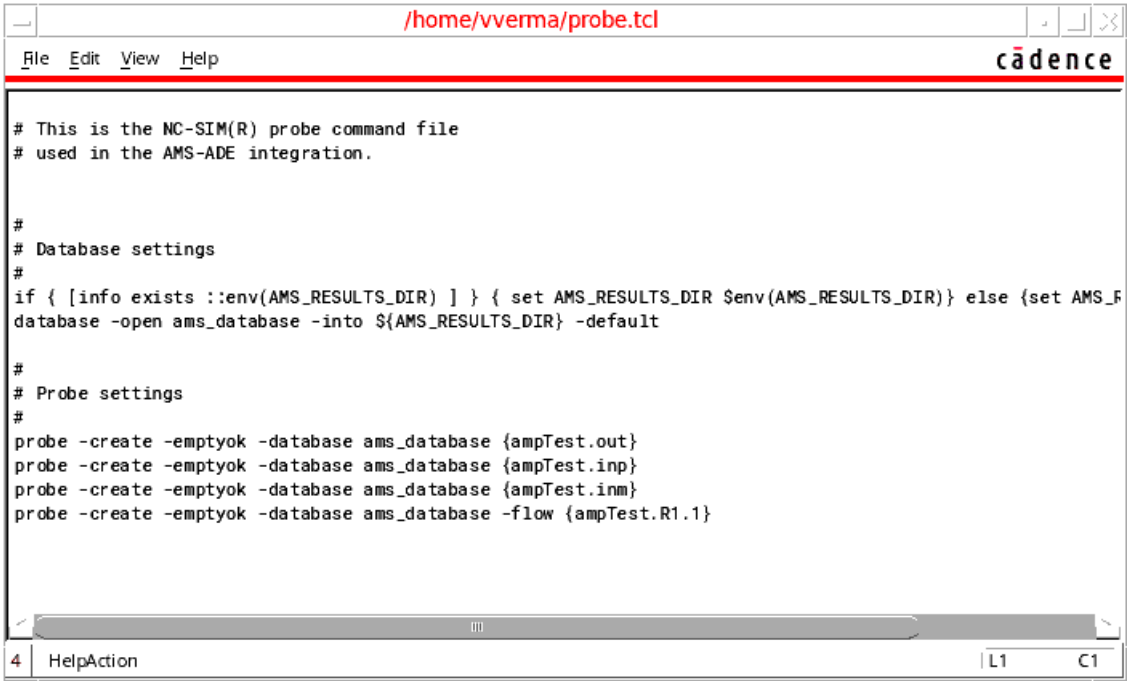




© 2023 Cadence Design Systems, Inc.

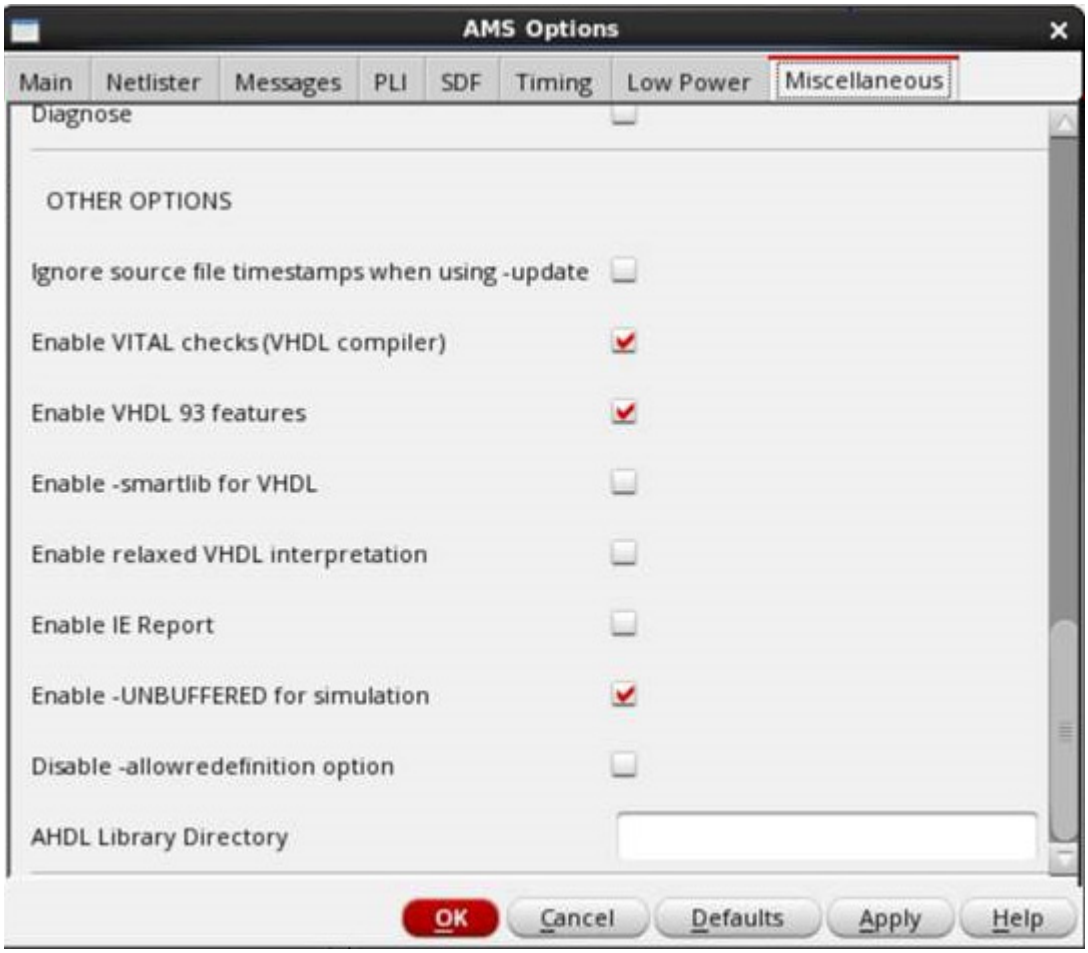
- The *View Probe Tcl file* button helps you view the probe and xmsim Tcl options, such as setting assertion commands, before running the simulation.

  When you select a Tcl file in the *Probe Tcl file* field, the following window appears:




- The *Enable Maestro Tcl file* check box enables editing the custom maestro Tcl file. If you select `on`, the *Edit Maestro Tcl file* button is enabled and allows you to open Virtuoso editor to edit `maestroDo.TCL` file. This file is saved at `lib/cell/view` directory. If you select `off`, the *Edit Maestro Tcl file* button is disabled.


© 2023 Cadence Design Systems, Inc.




In the *Miscellaneous* tab, you can specify INCLUDE options used by *xmvlog*, ACCESS options used by *xmelab* and PROFILER options used by *xmsim*. You can also specify few other VHDL options used by *xmvhdl*.

You can select the *Linter check* option to turn on the AHDL linter feature that enables you to detect modeling issues in analog/mixed-signal Hardware Description Languages (AHDL). The AHDL linter feature comprises of static and dynamic lint checks. Static lint checks are performed before analysis. Dynamic lint checks are performed during analysis for dynamic modeling issues. Choose *Simulation – Linter Log* in the simulation window to view the AHDL Linter report.

For more information about the AHDL Linter feature, refer to the description of AHDL Linter in the *Spectre AMS Designer and Xcelium Simulator Mixed-Signal User Guide*.

You can select custom AHDL library by specifying the directory path in *AHDL Library Directory.* If it is left empty, the default `-ahdllibdir` value is used.


© 2023 Cadence Design Systems, Inc.

## Additional Arguments Form

The Additional Arguments form lets you add additional arguments that cannot be specified in the AMS Options form.

| Field/Column | Description |
|---|---|
| ***Additional Arguments*** | This table lets you specify additional arguments that cannot be specified in the AMS Options form. |
| *Enable* | Enables or disables the specified argument. |
| *Argument Type* | Specifies whether the given argument is a Spectre or xrun argument. |
| *Additional Argument* | Lets you specify additional arguments that cannot be specified in the AMS Options form. |
| *Pre-commands for xrun* | Controls the availability of the fields *KSH Commands being included* and *KSH File including commands*.<br>■ `none`: Disables the fields *KSH Commands being included* and *KSH File including commands*.<br>This is the default.<br>■ `command`: Enables the field *KSH Commands being included*.<br>■ `file`: Enables the field *KSH File including commands*. |
| *KSH Commands being included* | Specifies the ksh commands to be added to the `runsimulation` file, before the `xrun -f` command. |
| *KSH File including commands* | Specifies a file containing Ksh commands that are then copied to the `runSimulation` file, before the `xrun -f` command. |

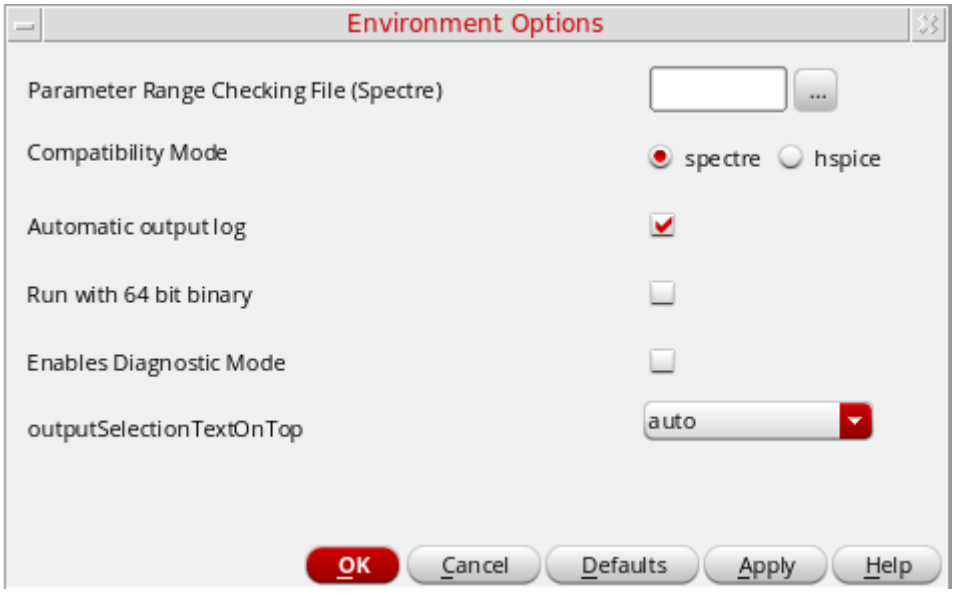

© 2023 Cadence Design Systems, Inc.

## Setting up Environment Options

To set the environment options for the AMS simulator:

➡ Choose *Setup – Environment*.

The Environment Options form appears.




For information about these options, see Setting up Simulator Environment Options.

Video

Configuring IE card information using IE card setup in ADE Explorer

### *Setting IE Cards*

The interface element (IE) card specifies the parameters, optionally, for a particular design unit. If you do not specify a design unit, the elaborator applies the parameter settings to all IEs globally.

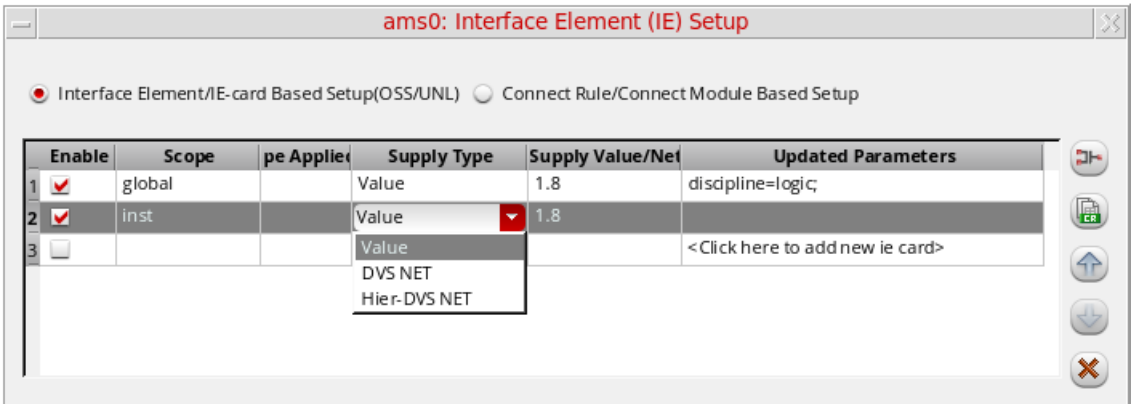
You can use an IE card to automate the process of creating a custom discipline and connect rule for connecting the custom discipline to the electrical discipline. The software applies the custom discipline to domainless nets in your design. So, if you have digital modules with undeclared port disciplines, you can use an IE card to specify a discrete discipline for domainless nets and the elaborator will insert the appropriate connect module automatically.

To set the IE card parameters:


© 2023 Cadence Design Systems, Inc.

1. In the ADE Explorer window, choose *Setup – Connect Rules/IE Setup.*

   The *Interface Element (IE) Setup* form appears.




   The *Interface Element (IE) Setup* form enables you to add an interface element and its parameters. Each row in the table contains the following information:

   - *Enable*: If selected, adds the IE card information in the `amsd` control block. You can select the check box in the *Enable* column to add an IE card.
   - *Scope*: Specifies the scope for the IE card. By default, *global* is selected as the scope. However, you can select the following scopes from the drop-down list: *lib*, *cell*, *celluport*, *cellport*, *inst*, *instport*, *net*.
   - *Scope Applied To*: Specifies the name of the instance/port/net/library/cellname, based on value selected from the *Scope* drop-down list, to which the scope is applied. You can either type a value in this field or click the table cell and then click the browse button ( ) to select the instance/port/net/library/cellname from the schematic or Library Manager. Alternatively, you can click the schematic button( ) to select a value from the schematic or LibraryManager. This field is editable and can also accept wild cards to specify designated multiple instances/ nets/ports. You can specify multiple entries with each entry separated by a semicolon (;) as the delimiter.

     **Note:** All scope types, except *global*, require a value to be specified in the *Scope Applied To* field to generate the IE card.
   - *Supply Type*: Specifies the voltage power supply type, based on the value selected from the Supply Type drop-down list, namely, *Value*, *DVS NET* and *Hier-DVS NET*. If you set this field to *Value*, then the *Supply Value/Net* field will accept a Supply voltage value. If you set this field to *Hier-DVS NET* then the *Supply Value/ Net* field will accept a hierarchical path to a net as a value.


© 2023 Cadence Design Systems, Inc.

- *Supply Value/Net*: Specifies the voltage power supply value. You can either specify a constant value, a design variable by using the VAR syntax, or type the hierarchical path to a net in this field. Alternatively, you can click the table cell and then click the browse button ( ) to select the net from the schematic. A net selected from the schematic automatically becomes a VDD net. Based on the value that you specify on this field, the *connrule* parameter value in the *Extended* tab automatically changes to *CR_full_fast* versus *CR_dynsup_full_fast* and the respective supply parameter receives the value `vsup` or `vddnet` on the respecitive connect rules.

  **Note:** The *CR_dynsup_full_fast* connect rule is available in INCISIV 15.1 or a later release. Therefore, to access dynamic voltage supply feature, you need to set the appropriate INCISIV release.

  For example, a numeric value or design variable leads to the selection of the *CR_full_fast* connect rule, whereas a hierarchical net will lead to the selection of the *CR_dynsup_full_fast* connect rule.

  **Note:** When the *CR_dynsup_full_fast* connect rule is selected, you must specify a value for the mandatory parameter, *vssnet,* which is displayed on the *Basic* tab.

  You must specify a value in the *Supply Value/Net* field whenever a connect rule associated with selected IE card needs voltage supply.

  **Note:** When the `CR_dynsup_full_fast` connect rule is selected, the *Supply Value/Net* field only accepts a hierarchical path to a net as a value.

  The default connect rule for the IE card is *CR_full_fast*, which has `vsup` as the supply parameter with a default value of `1.8V`. Therefore, by default, `1.8V` is specified for this field. There are some other connect rules for which `vsup` is not required. For such connect rules, this field does not contain any value.

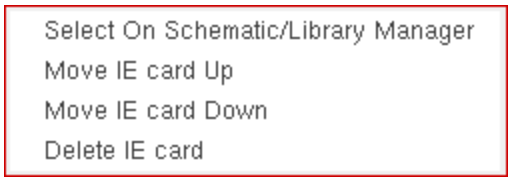
  If you have two IE cards that have the same discipline, you must specify the same supply voltage for them.

- *Updated Parameters*: Displays the IE card parameters that have been updated.

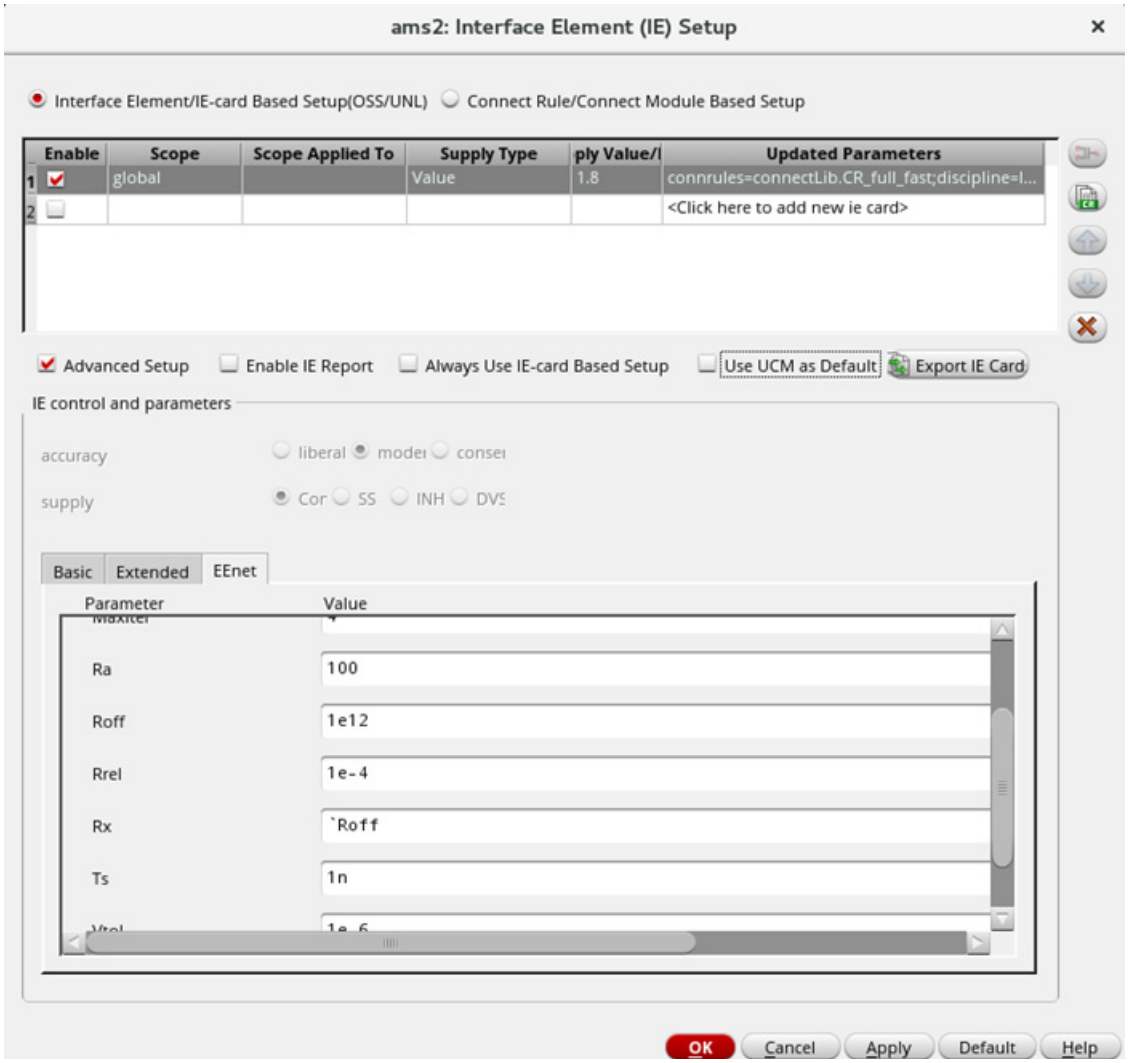
  You can move an IE card up or down the table by clicking the up ( ) and down ( ) arrow buttons. You can view the connect rule associated with the selected IE card by clicking ( ). You can delete an IE card by clicking the delete ( ) button.


© 2023 Cadence Design Systems, Inc.

You can also select some of the above options by clicking the right mouse button on the selected IE card and selecting the required option from the context menu, as shown below.



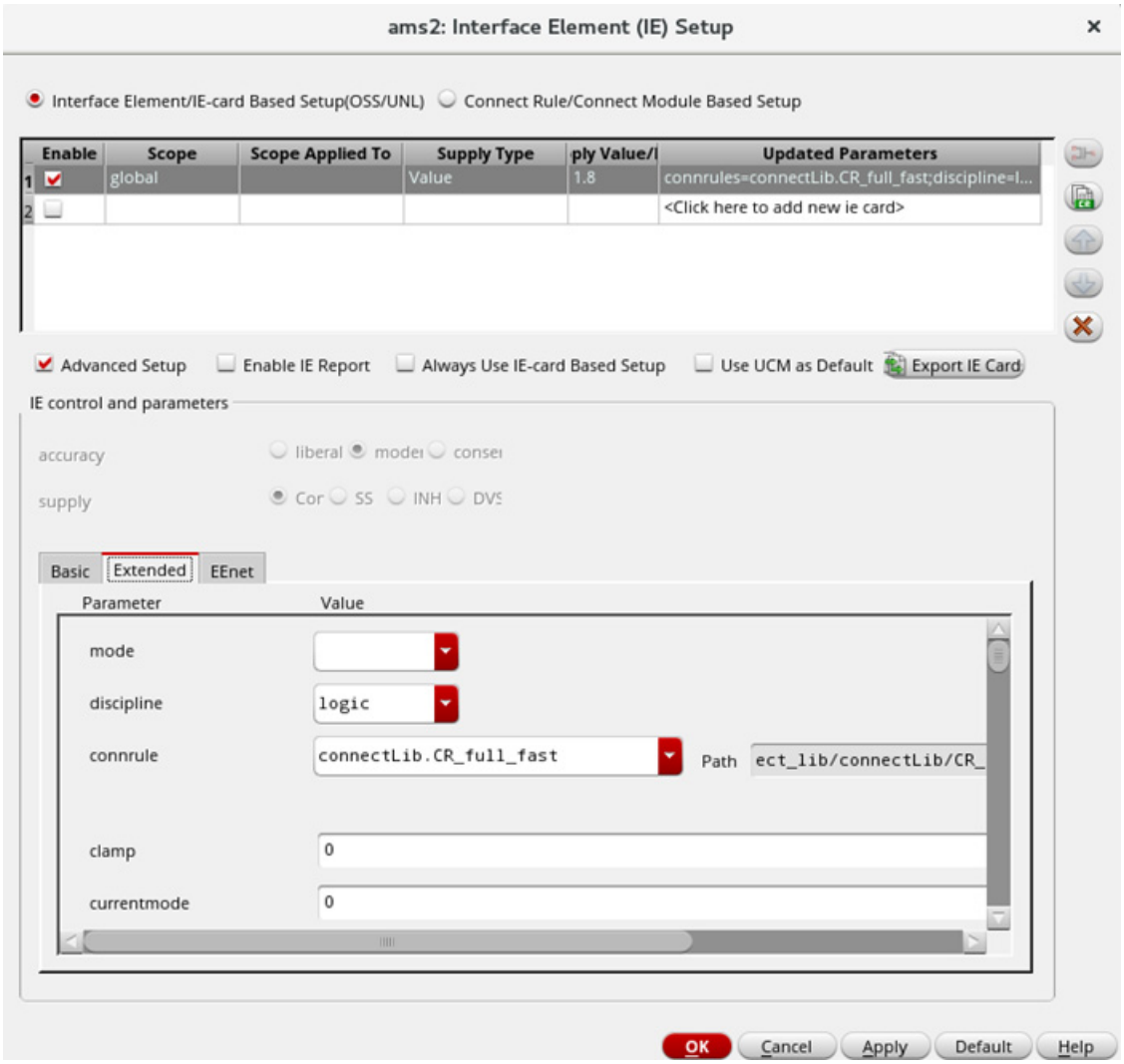

You can select the *Advanced Setup* option to change the IE card parameters for the selected scope. When you select the *Advanced Setup* check box, the *Interface Element (IE) Setup* form expands to display the IE card parameters that you can modify for the selected IE card in the IE cards table.





© 2023 Cadence Design Systems, Inc.

The *IE control and parameters* section consists of three tabs, *Basic*, *Extended*, and *EEnet* listing the parameters with their default values. You can modify a parameter by changing the value as per your requirements. You can place the mouse pointer over a parameter name to display a tooltip that provides a brief description about the parameter and its default value in the associated connect rule, which can be changed using the *Extended* tab.




In the *Extended* tab, you can set the following parameters:

- *mode* - Enables you to insert the interface element as splitted or merged.
- *connrule* - Enables you to select a connect rule for the associated IE card in the IE cards table from the drop-down list. The default connect rule is *CR_full_fast*.


© 2023 Cadence Design Systems, Inc.

Select the *<Using UCM>* option to set up IE cards using the Universal Connect Module (UCM) and to view the options *accuracy* and *supply*.

- *discipline*: Enables you to select the discrete discipline associated with the automatically chosen connect module from the selected connect rule. Default is an empty discipline which causes xmelab to automatically generate a discipline, however, if you have your own discipline then you can specify it in this field. The drop-down list contains built-in disciplines in addition to any other predefined disciplines that are specified using the `.cdsinit disciplinesList` variable. For example, `envSetVal("ams.envOpts" "disciplinesList" 'string "discipline1 discipline2")` in the `.cdsinit` file will add `discipline1` and `discipline2` along with the built-in disciplines in the *discipline* drop-down list.

  In addition, you can change other IE parameters such as `vto`, `rz`, `vdelta`, `vtoldelta` and so on.

- *Enable IE Report*: Generates a detailed IE report containing information about IE, port discipline, sensitivity, and so on. This option is also presented under the *Miscellaneous* tab of the *AMS Options* form. You can choose *Simulation – Output Log – IE Report i*n the ADE Explorer window to view the IE report.

- *Always Use IE-card Based Setup*: Disables the options *Interface Element/IE-card Based Setup(OSS/UNL)* and *Connect Rule/Connect Module Based Setup* and displays only the options available for IE-card based setups.

- *Use UCM as Default*: Specifies that by default, any new IE cards must be inserted using the UCM. UCM consolidates and replaces the existing connect modules in `connectLib` in the Xcelium installation. The IE cards are then created with the default UCM configuration unless the parameter values are specified explicitly. This means that by default, the values for the parameters `accuracy` and `supply` are set to `moderate` and `Const/LPS` respectively.

  Note the following:

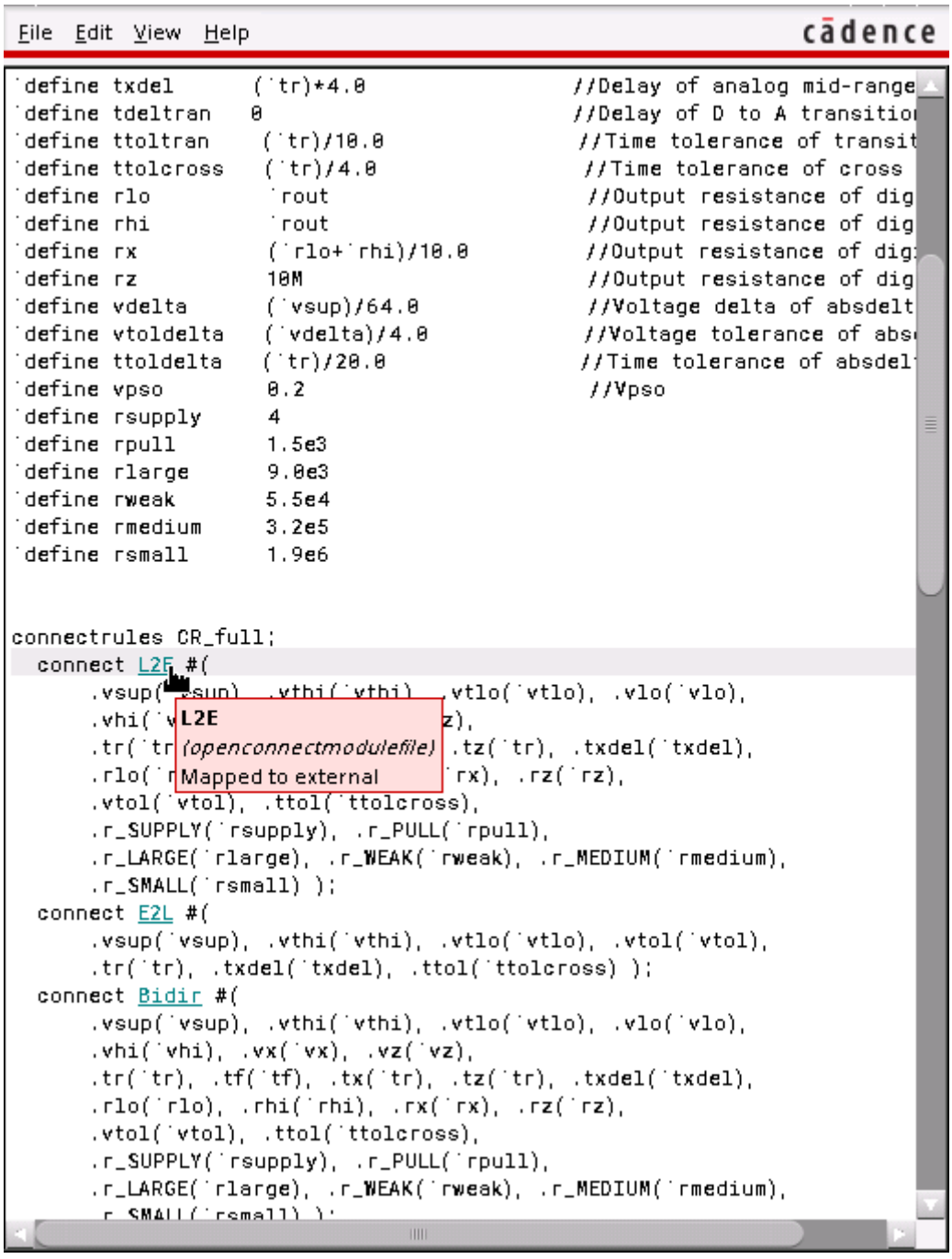
  - This check box is selected automatically when the simulation performance mode is set to Spectre FX.
  - Selecting this check box does not affect the existing IE cards in the table.
  - Cadence recommends that you enable this feature by using the environment variable ams.envOpts ieUseUcmAsDefault while setting up a maestro session.

Click the *Default* button to change the values of all IE parameters back to their default values.

Click the *View Connect Rule File* button to open the connect rule file.


© 2023 Cadence Design Systems, Inc.

The connect rule file opens in a text editor. Click any hyperlink provided in the connect rules file to view the associated connect module file, as shown below:




The connect module file opens in a text editor.


© 2023 Cadence Design Systems, Inc.

### *connectRules_newIE.il file*

The `connectRules_newIE.il` file contains the built-in connect rules which are displayed in the *connrule* drop-down list box.

ADE Explorer searches for the `connectRules_newIE.il` file in the following locations:

- All libraries defined in the `cds.lib` file
- Path specified by the c`onnectRulesPath` variable in the .cdsenv file.

If multiple `connectRules_newIE.il` files are located, their contents are concatenated and displayed in the Interface Element (IE) Setup form.

All Cadence-supplied connect rules or connect modules are stored in the `connectRules_newIE.il` file located in the `/tools/affirma_ams/etc/connect_lib/connectLib` directory in your Cadence Incisive Unified Simulator (IUS) installation directory.

You can use the `genIeConnRulesFile` command to invoke automatic compilation of your user-defined connect rules into a custom `connectRules_newIE.il` file.

```
genIeConnRulesFile [-help] [-destpath <destination path>] -lib <lib1> <file1>
[file2 ...] [-lib <lib2> <file3> [file4 ...]] [-rulefile <rulesFile>] [-rule
<ruleName1> [ruleName2 ...]]
```

Here:

- `-help` displays the help for the `genIeConnRulesFile` command.
- `-destpath <`*`destination path`*`>` specifies the directory in which the `connectRules_newIE.il` file will be created.
- `-lib <`*`lib1`*`> <`*`file1`*`>` [`<`*`file2`*`>` ...] specifies the libraries in which the connect rule files exist and the connect rule files that need be parsed. All the connect rules in the specified rule files are included in the `connectRules_newIE.il` file.

  For example, to include all the connect rules in the `crules1.vams` and `crules2.vams` rule files that exist in the `mylib` library, use the following command:

  ```
  genIeConnRulesFile -destpath . -lib mylib mylib/crules1.vams mylib/
  crules2.vams
  ```

  In this example, the `connectRules_newIE.il` file is created in the current directory.

  **Note:** If you use the `-rulefile` or `-rule` option, only the rules specified by the `-rulefile` or `-rule` option will be included in the `connectRules_newIE.il` file.
- `-rulefile <rulesFile>` specifies the name of a file which contains the list of connect rules that need to be included in the `connectRules_newIE.il` file.


© 2023 Cadence Design Systems, Inc.

For example, to include only the connect rules named `cr1`, `cr2`, `cr8` and `cr9` in the `connectRules_newIE.il` file, do the following:

**a.** Create a file named `myrules` with the following text:

```
cr1
cr2
cr8
cr9
```

**b.** Use the following command:

```
genIeConnRulesFile -destpath . -lib mylib mylib/crules1.vams mylib/
crules2.vams -rulefile mylib/myrules
```

In this example, even though the `-lib` option specifies the `crules1.vams` and `crules2.vams` rule files, only the rules listed in the `myrules` file are included in the `connectRules_newIE.il` file.

- `-rule <ruleName1> [<ruleName2> ...]` specifies the names of the rules in the specified rule files that need to be included in the `connectRules_newIE.il` file.

  For example, to include only the connect rules `cr1` and `cr2` in the `crules1.vams` rules file and the connect rules `cr8` and `cr9` in the `crules2.vams` rules file, use the following command:

  ```
  genIeConnRulesFile -destpath . -lib mylib mylib/crules1.vams mylib/
  crules2.vams -rule cr1 cr2 cr8 cr9
  ```

  In this example, even though the `-lib` option specifies the `crules1.vams` and `crules2.vams` rule files, only the rules specified using the `-rule` option are included in the `connectRules_newIE.il` file.

When importing user-defined connect rules from a connect rule file or a pre-compiled connect library, ADE compiles the connect rule file and generates the `connectRules_newIE.il` file under the `/tmp` directory and then it merges the imported connect rules with the existing connect rules in the IE Setup form. The connect rule file from pre-compiled connect library is searched under the selected library, cell, and view.

**A few points to note**:

- The parameters specified in the Basic parameters section in the connect rule file, which is the section below the `//Basic parameters` statement, appear in the *Basic* tab of the Interface Element (IE) Setup form.
- The parameters specified in the Extended parameters section in the connect rule file, which is the section below the `//Extended parameters` statement, appear in the *Extended* tab of the Interface Element (IE) Setup form.


© 2023 Cadence Design Systems, Inc.

- The parameters that are not defined in either of the two sections, Basic parameters or Extended parameters, are parsed and merged in the *Basic* and *Extended* tab.
- The words `vsupmin`, `vsup`, `tr`, `rout`, `vddnet`, and `vssnet` are considered as keywords. If any of these words are present in the file, they are displayed in the *Basic* tab. All the other parameters are displayed in the *Extended* tab.
- Parameters in both the tabs, *Basic* and *Extended*, are arranged in alphabetical order.
- Parameter names are not case sensitive.
- If a parameter is defined with the same name more can once, then the last specified value of the parameter is used.

Video

For an overview of IE Card Setup, see the Configuring IE Card Information Using IE Card Setup in ADE Explorer demonstration available on Cadence Online Support.

### *Setting Connect Rules*

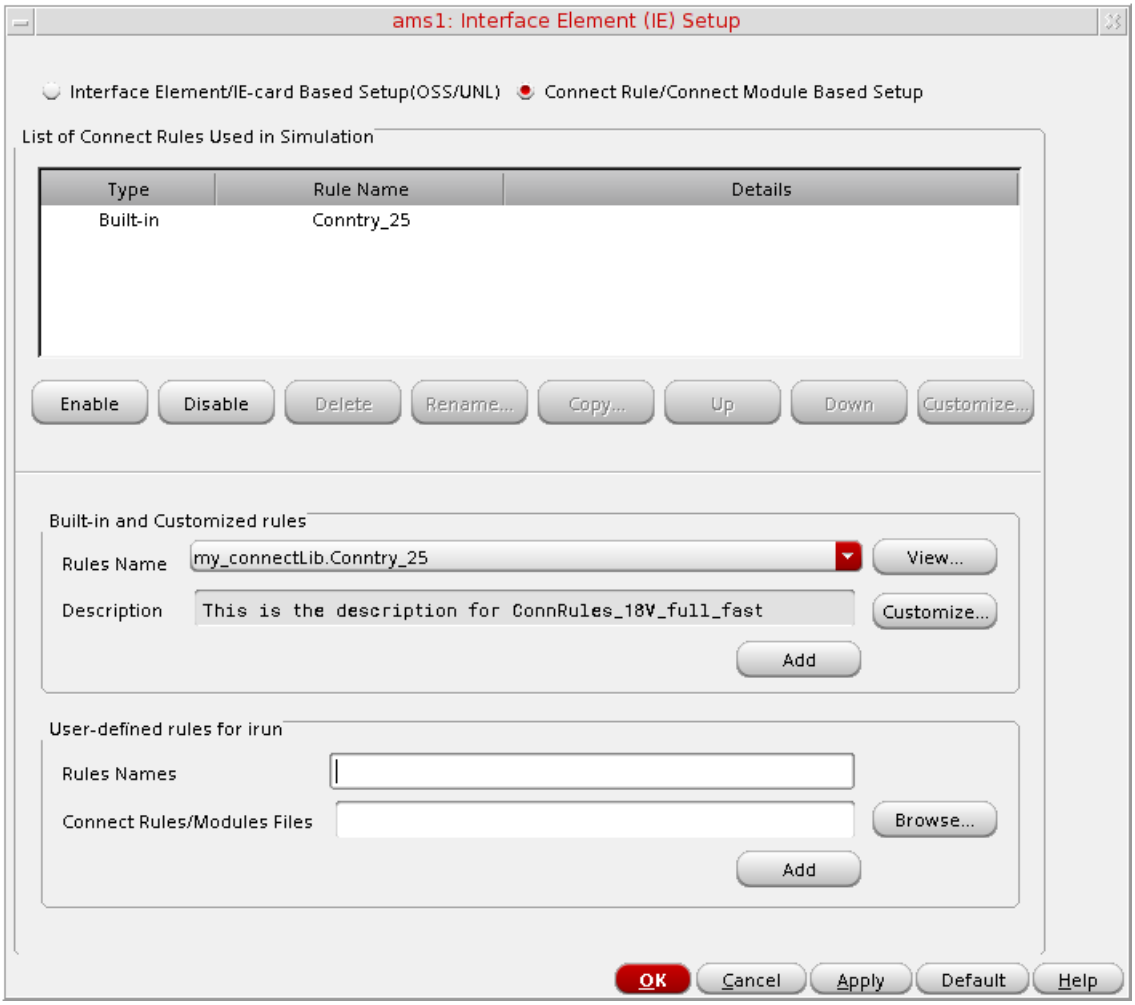
A connect rule specification is used to insert selected connect modules within a mixed signal scope. A connect module is a module that is automatically inserted to connect the continuous and discrete domains through disciplines (mixed-signal nets) of the design hierarchy, together. For more information, see the *Connect Modules* section in the “Mixed-Signal Aspects of Verilog-AMS” chapter of the *Cadence Verilog-AMS Language Reference*.

To set connect rules,

1. In the ADE Explorer window, choose *Setup – Connect Rules/IE Setup.*


© 2023 Cadence Design Systems, Inc.

The *Interface Element (IE) Setup* form appears.




2. Choose the *Connect Rule/Connect Module Based Setup* option.

   The form displays a list of connect rules used in the simulation that are passed to the elaborator. For using the default set of connect rules, you can set an environment variable called `connectRulesList` in .cdsinit.

   - The *List of Connect Rules Used in Simulation* table displays the connect rules that will be used when the run mode is *AMS Unified Netlister with xrun*. For more information about the *AMS Unified Netlister with xrun* run mode, see Setting Up the AMS Netlister Options on page 1112.

     Each row in the table has the following details:

     - *Type*—Displays the type of the connect rule as `Built-in`, `User-defined` or `Modified built-in`.


© 2023 Cadence Design Systems, Inc.

- *Rules Name* —Displays the name of the rule, which is the cell name.
- *Details*—Displays the name of the library in which the rule exists and the view name for the rule.

**Note:** You can use the `genConnRulesFile` command utility to have your user-defined rules displayed as built-in rules from the *Built-in rules* group box. For more information about the `genConnRulesFile` command, see connectRules.il File on page 1060.

### *Adding Connect Rules*

You can add built-in or user-defined connect rules. To add a built-in connect rule, choose the built-in connect rule from the *Rules Name* drop-down list and click *Add*.


© 2023 Cadence Design Systems, Inc.

- If you want to view the rule, click the *View* button to open the connect rules file in a text editor.

- If you want to customize the rule, click the *Customize* button.


© 2023 Cadence Design Systems, Inc.

For more information about customizing built-in rules, see Customizing Built-in Rules on page 1062.

**Note:** When you customize a built-in rule, its type appears as *Modified built-in* and a number is suffixed to its name. For example, a built-in rule `ConnRules_3V_mid`, when customized, is renamed to `ConnRules_3V_mid1`.

- Click the *Add* button to add the rule.
- To add a user-defined connect rule to be used when the run mode is *AMS Unified Netlister with xrun*, do the following:
  - In the *Rules Names* field, enter the name of the connect rule to be used by xrun.
  - In the *Connect Rules/Modules Files* field, enter the filename and path of the connect rule and connect module files in which the specified rule and used connect module exist. You can use the *Browse* button to select the file and click *Open* to add them in the field.

    The connect rule is displayed in the *List of Connect Rules Used in Simulation* table.

    You can also use environment variables to specify the path to the file. For example, you can specify the path as:

    ```
    $CONNECT_FILES/myrules.vams
    ```

  Note the following:

  - If you try to enter more than one rule at a time, you will get a pop-up error saying that you can enter only one rule name at a time.
  - If a rule with the same name exists in more than one connect rule file, the rule that exists in the first connect rule file displayed in the *Connect Rules/ Modules Files* field will be used.
  - If you do not specify a rule name in the *Rules Names* field, the first rule in the first connect rule file specified in the *Connect Rules/Modules Files* field will be used.
  - The user-defined connect rules specified in the *Rules Names* field will not be displayed in the *List of Connect Rules Used in Simulation* table.
- The built-in connect rules displayed in the *List of Connect Rules Used in Simulation* table and the connect rules specified in the *User-defined rules for xrun* group box will be used in the simulation. So you can have a mixture of both built-in and user-defined rules in one simulation.

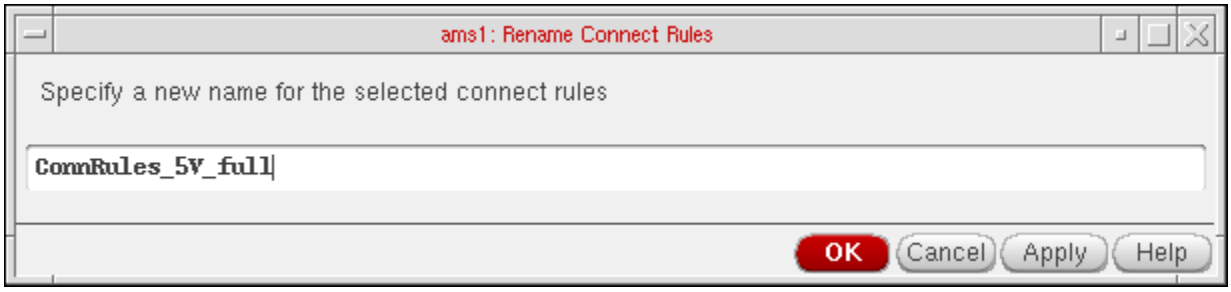

© 2023 Cadence Design Systems, Inc.

### *Renaming Connect Rules*

To rename a rule in the *List of Connect Rules Used in Simulation* table,

1. Select a rule from the table.
2. Click the *Rename* button.

   The Rename Connect Rules pop-up box appears.




3. Specify a unique name for the selected connect rule and click *OK* or *Apply*.

   If a rule by the same name exists in the Connect Rules table, an error message appears and the pop-up remains open.

   The list of rules in the Select Connect Rules from also shows the modified connect rule name.

### *Deleting Connect Rules*

To delete a rule in the *List of Connect Rules Used in Simulation* table,

1. Select one or more rules in the *List of Connect Rules Used in Simulation* table.
2. Click the *Delete* button.

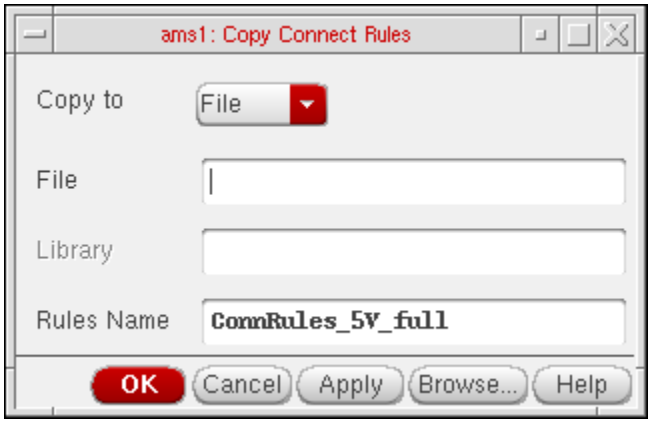
### *Copying Connect Rules*

To copy a rule in the *List of Connect Rules Used in Simulation* table,

1. Select a rule in the *List of Connect Rules Used in Simulation* table.
2. Click the *Copy* button.


© 2023 Cadence Design Systems, Inc.

The *Copy Connect Rules* form appears.




3. Select either *Library* or *File* in the *Copy to* pull-down box to indicate where you want the rule copied to.

   If your choice is *Library*, the *Library* and *Rules Name* fields are enabled and you need to specify the relevant values in them. You may either type in these values or select them using the *Browse* button. If your choice is *File*, the *File* field is enabled and you need to specify a filename for the rule to be copied into. You can specify a filename indicating the path. If you specify a filename without a path, it implies that it is in the current working directory. If you specify a name for a file on which you do not have read permission, you will see an error message saying so.

   The Browse button brings up the library browser if you select *Library* and the file browser if you select *File*.

4. Click *OK*.

   The new copied rule appears in the connect rules table. This is a convenient way to copy your modified connect rule to a permanent location in your library. The connect rules will be saved in your state files, but you may want to save it to a permanent library as well.

### *Changing the Sequence of Connect Rules*

To change the sequence of the rules displayed in the *List of Connect Rules Used in Simulation* table,

1. Select a rule in the *List of Connect Rules Used in Simulation* table.
2. Click the *Up* or *Down* buttons.

   **Note:** All the selected rules are auto-compiled. They are passed to the elaborator for processing. If two rows have a matching statement, the first of these rows is used. If two


© 2023 Cadence Design Systems, Inc.

rules in a particular row have a matching statement, the last of these statements is used.

3. Click *OK* to save the settings in the current session.

### connectRules.il File

The `connectRules.il` file contains the built-in connect rules that were specified when running the `genConnRulesFile` and are displayed in the Select Connect Rules form.

ADE Explorer searches for the `connectRules.il` file in the following locations:

- All libraries defined in the `cds.lib` file
- Current work directory
- `$HOME` directory
- `$CDS/share/cdssetup/ams/`
- Path specified by the `connectRulesPath` variable in the `.cdsenv` file

If multiple `connectRules.il` files are located, their contents are concatenated and displayed in the Select Connect Rules form.

All Cadence-supplied connect rules/ connect modules are stored in the `connectRules.il` file located in the `/tools/affirma_ams/etc/connect_lib/connectLib` directory in your Cadence Incisive Unified Simulator (IUS) installation directory.

You can use the `genConnRulesFile` command to invoke automatic compilation of your user-defined connect rules into a custom `connectRules.il` file.

```
genConnRulesFile [-help] [-destpath <destination path>] -lib <lib1> <file1>
[file2 ...] [-lib <lib2> <file3> [file4 ...]] [-rulefile <rulesFile>] [-rule
<ruleName1> [ruleName2 ...]]
```

where:

- `-help` displays the help for the `genConnRulesFile` command.
- `-destpath <`*`destination path`*`>` specifies the directory in which the `connectRules.il` file will be created.
- `-lib <`*`lib1`*`> <`*`file1`*`>` [`<`*`file2`*`>` ...] specifies the libraries in which the connect rule files exist and the connect rule files that need be parsed. All the connect rules in the specified rule files are included in the `connectRules.il` file.

  For example, to include all the connect rules in the `crules1.vams` and `crules2.vams` rule files that exist in the `mylib` library, use the following command:


© 2023 Cadence Design Systems, Inc.

```
genConnRulesFile -destpath . -lib mylib mylib/crules1.vams mylib/crules2.vams
```

In this example, the `connectRules.il` file is created in the current directory.

**Note:** If you use the `-rulefile` or `-rule` option, only the rules specified by the `-rulefile` or `-rule` option will be included in the `connectRules.il` file.

- `-rulefile <rulesFile>` specifies the name of a file which contains the list of connect rules that need to be included in the `connectRules.il` file.

  For example, to include only the connect rules named `cr1`, `cr2`, `cr8`, and `cr9` in the `connectRules.il` file, do the following:

  **a.** Create a file named `myrules` with the following text:

  ```
  cr1
  cr2
  cr8
  cr9
  ```

  **b.** Use the following command:

  ```
  genConnRulesFile -destpath . -lib mylib mylib/crules1.vams mylib/
  crules2.vams -rulefile mylib/myrules
  ```

  In this example, even though the `-lib` option specifies the `crules1.vams` and `crules2.vams` rule files, only the rules listed in the `myrules` file are included in the `connectRules.il` file.

- `-rule <ruleName1> [<ruleName2> ...]` specifies the names of the rules in the specified rule files that need to be included in the `connectRules.il` file.

  For example, to include only the connect rules `cr1` and `cr2` in the `crules1.vams` rules file and the connect rules `cr8` and `cr9` in the `crules2.vams` rules file, use the following command:

  ```
  genConnRulesFile -destpath . -lib mylib mylib/crules1.vams mylib/crules2.vams
  -rule cr1 cr2 cr8 cr9
  ```

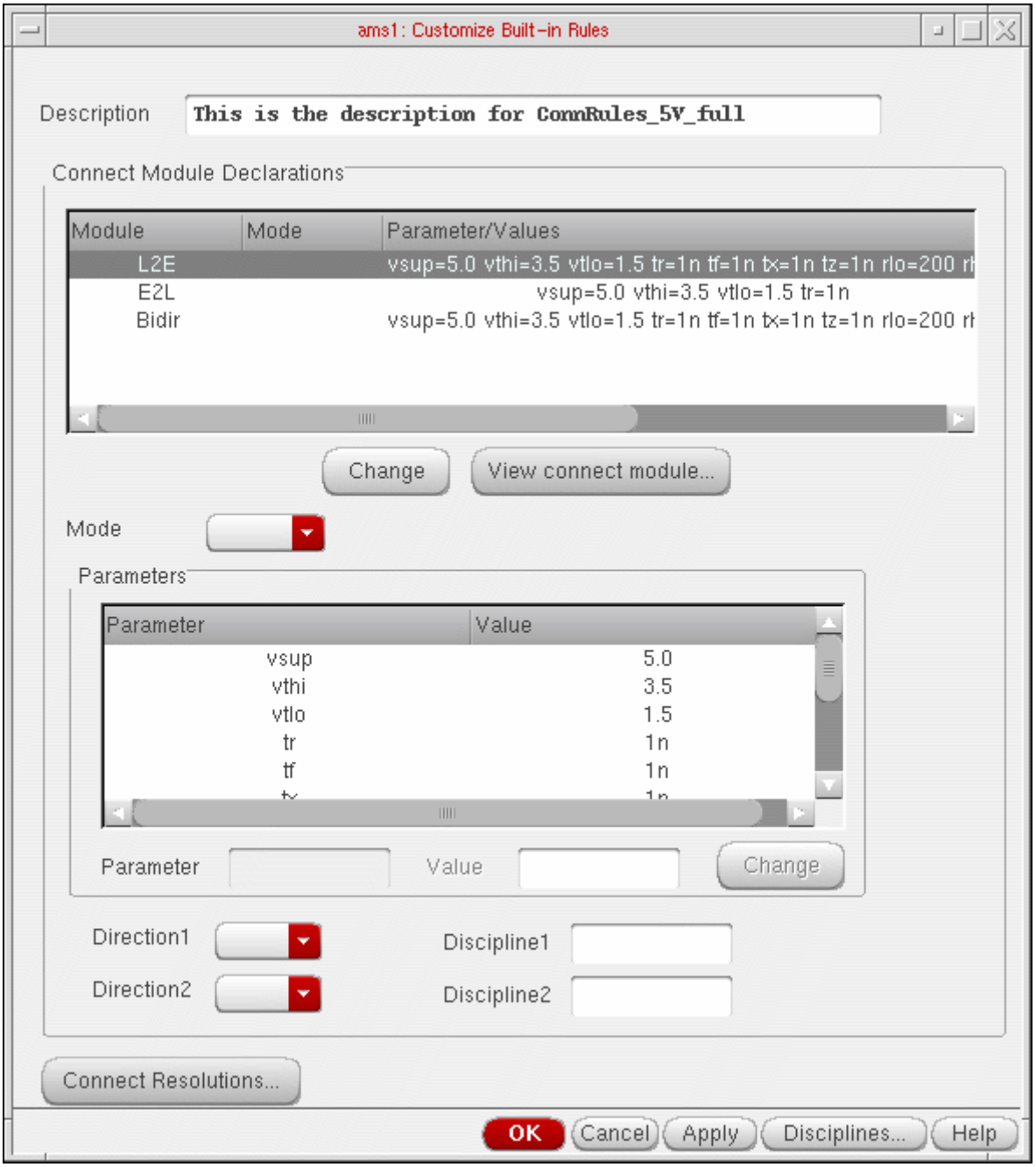
  In this example, even though the `-lib` option specifies the `crules1.vams` and `crules2.vams` rule files, only the rules specified using the `-rule` option are included in the `connectRules.il` file.

If you place your custom `connectRules.il` files in one of the locations where ADE Explorer searches for the `connectRules.il` files, the connect rules defined in the custom files are also displayed as built-in rules in the Select Connect Rules form.


© 2023 Cadence Design Systems, Inc.

## Customizing Built-in Rules

This form appears when you click the *Customize* button in the Select Connect Rules form.




You use this form as follows:


© 2023 Cadence Design Systems, Inc.

1. Specify a description for the rule in the *Description* field. This replaces the contents of the corresponding non-editable field in the Select Connect Rules form.
2. The *Connect Module Declarations* table displays the following information about the modules in the selected connect rule:
   - *Module* shows the name of a connect module.
   - *Mode* can be blank or have either of the values `merged` or `split`. When it is blank, it indicates that there is only one port of discrete discipline on the signal. The `split` value indicates that there should be one connect module inserted for each port. The `merged` value, which is the default, specifies that only one connect module should be inserted for all the ports on a signal.
   - *Parameter/Values* shows a list of parameter values.
   - The direction of the first port appears in the *Direction1* column and for the second port in the *Direction2* column. These columns can be blank or have one of these values: `input`, `output` or `inout`.

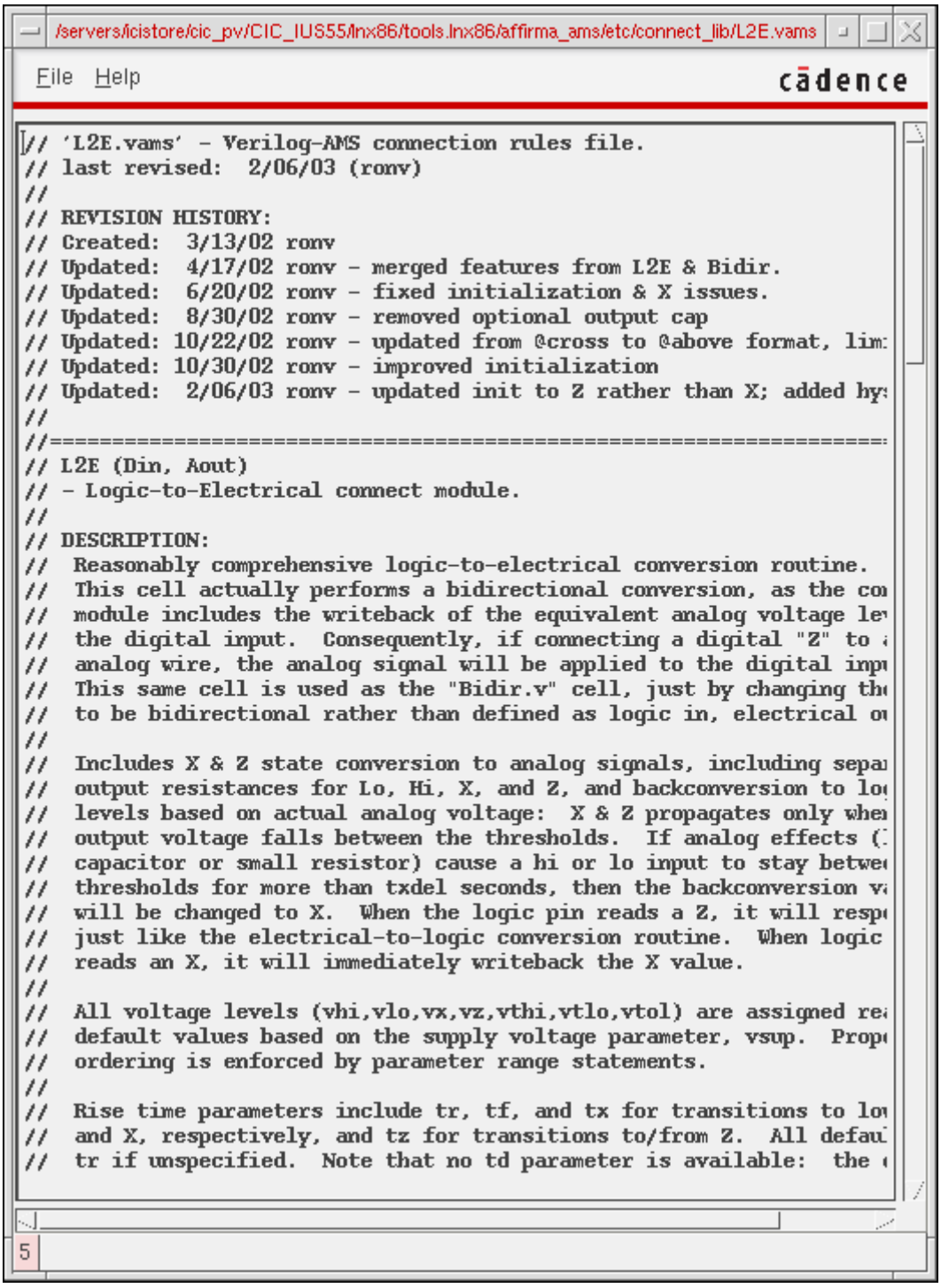
     You may need to scroll to the right to see these and the remaining columns.
   - The discipline information for the first port appears in the *Discipline1* column and for the second port under *Discipline2*. These columns may be blank for modules that do not have them specified.

   Select a module by clicking on the related row.
3. When a module is selected, these fields get populated with the related values: *Mode*, *Direction1*, *Direction2*, *Discipline1*, and *Discipline2*. After you modify one or more values, click the *Change* button just below the *Connect Module Declarations* table. The changes are reflected in the table. If you select multiple rows, this *Change* button appears disabled.

   When you select a row, the *Parameters* table is also populated. You can change values by selecting a row in this table, modifying the *Value* and clicking the *Change* button next to it.
4. If you want to see the connect module file, click the *View connect module* button.


© 2023 Cadence Design Systems, Inc.

This brings up the related connect modules file as shown below.




If a corresponding file does not exist, an error message appears.


© 2023 Cadence Design Systems, Inc.

5. When you select only one module in the *Connect Module Declarations* table, the related parameters and values are listed in the *Parameters* table. You can select a parameter and change its value by typing over it in the *Value* field and clicking the *Change* button next to it.

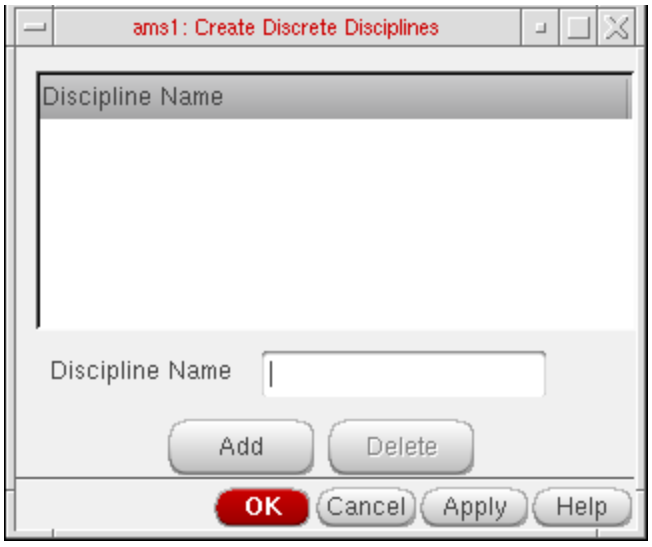
   When you select multiple rows in the *Connect Module Declarations* table, a union of all the parameters is shown in the *Parameters* table with their values. If a parameter has different values in different modules, its value is shown as blank. You can specify a value to be used for a particular parameter in the *Value* field and click the *Change* button next to it. The specified value applies to all the selected modules.

6. The *Direction1* and *Direction2* fields are blank by default. The possible combinations of values you may specify are:

   - *input*, *output*
   - *output*, *input*
   - *inout*, *inout*

7. The *Discipline1* and *Discipline2* fields are also blank by default. They may remain blank or you may specify a discipline-pair for the selected module.

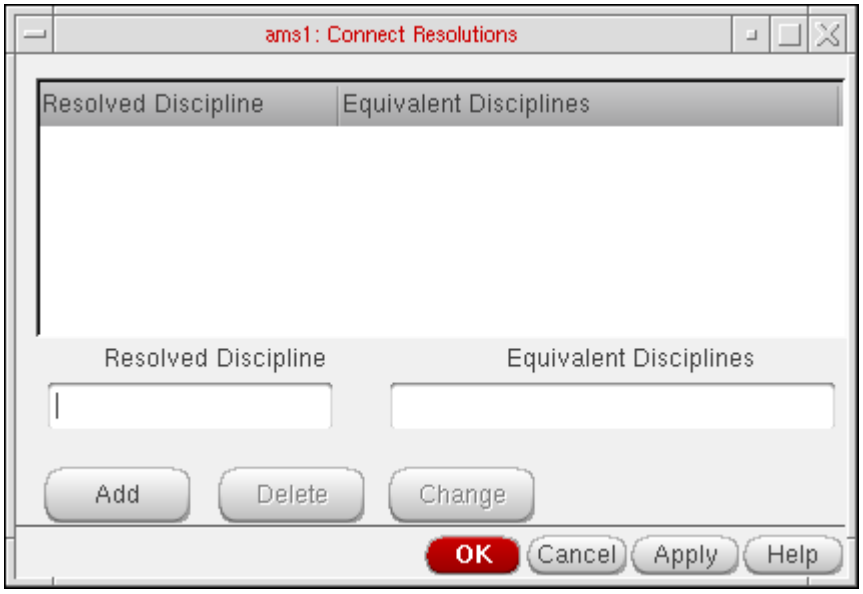
   You can also create discrete disciplines and specify them in these fields by clicking the *Disciplines* button, which brings up the *Create Discrete Disciplines* form.




   It shows a list of disciplines. You may type in a name in the *Discipline Name* field and click *Add* to create a new discipline. The name should be a legal verilog identifier. You may select one or more discipline names listed in the table and click the *Delete* button to delete them. This information is saved for the session when you click *OK* and is included in the connect modules information saved in a state file.


© 2023 Cadence Design Systems, Inc.

8. The *Connect Resolutions* button brings up the *Connect Resolutions* form.




It shows a list of *Resolved Disciplines* and *Equivalent Disciplines*. You can specify a discipline to be used when multiple nets with compatible disciplines are part of the same domain net.

- You can add a new association by specifying a *Resolved Discipline* and an *Equivalent Discipline* and clicking the Add button.
- You can modify an association by selecting a row, modifying the values in the *Resolved Discipline* and *Equivalent Discipline* fields and clicking the *Change* button.
- You can select one or more rows and delete them by clicking the *Delete* button.

9. Click the *OK* button to update the connect rules information with the changes made for connect resolutions.
10. Click the *OK* button in the Customize Built-in Rules form to update the connect rules information for the session.

### Using MATLAB/Simulink

The ability to run a cosimulation using MATLAB®/Simulink ®with AMS is now available. There are three use models to choose from:

- From ADE Explorer: You can start MATLAB from ADE Explorer.


© 2023 Cadence Design Systems, Inc.

- Separate: DFII/ADE Explorer is started and then MATLAB is separately started in standalone mode and then the two communicate for the co-simulation.
- From MATLAB: AMS is started from MATLAB via the runSimulation script.

**Note:** MATLAB and Simulink are registered trademarks of The MathWorks, Inc.

### Setting up the AMS/MATLAB Co-simulation

In order to cosimulate between AMS and Simulink®, coupler modules are required for both AMS and Simulink. For AMS, the coupler module, will be placed as part of your design in the schematic. Associated with each coupler is a verilog.vams file that contains the coupler module. This coupler module contains the system calls: `$couple_init` which calls the VPI code that will be used to communicate with Simulink. The system task `$set_access_readwrite` ensures that the proper read/write access is set for the coupler module.

There are two types of coupler cells to choose from: fixed cell coupler or Pcell coupler. The Pcell coupler is located in analogLib. For more information, see description for simulinkCoupler in Analog Library Reference. You can place the Pcell coupler and configure the number of input and output pins along with other options described below. You can create the matching `verilog.vams` file that contains the correct inputs/outputs as configured in the Pcell on the schematic by using the GUI in ADE Explorer as described below.

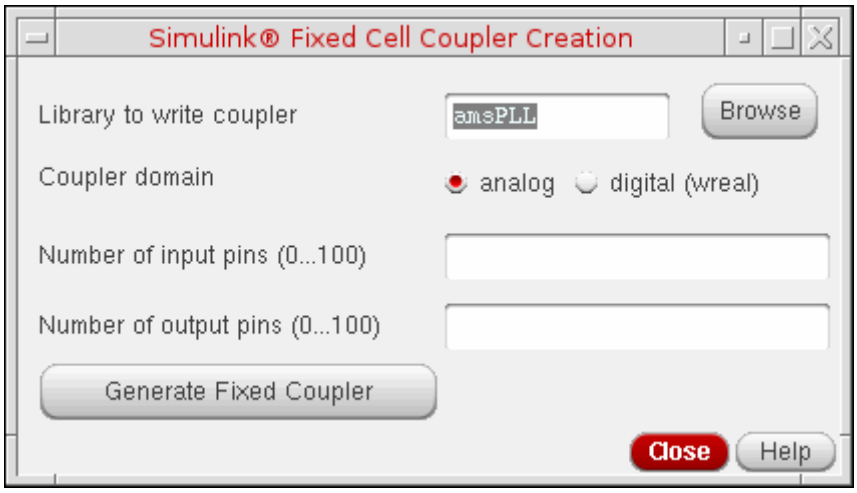
If there are specific input/output configurations that you know you will use frequently, you can create a fixed cell coupler and use that fixed cell in your schematic rather than a Pcell. The `verilog.vams` file will be created for the fixed cell when the fixed cell is created. To create a fixed cell coupler:

1. In the schematic, choose *Launch – Plugins – Mixed Signal Options – AMS*.

   This results in the AMS menu being added to the Composer banner.
2. Select *AMS - Simulink Coupler Creation*.


© 2023 Cadence Design Systems, Inc.

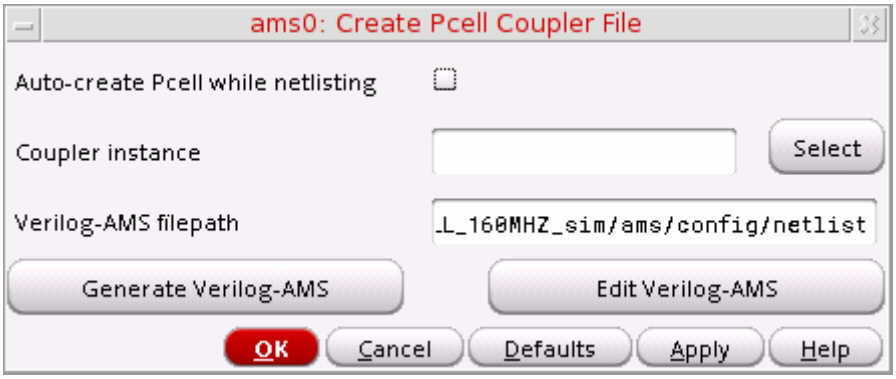
The Simulink Coupler Fixed Cell Creation form opens.




3. Enter the Library name where you want the fixed coupler to be placed. You may want to create a coupler library for all the fixed couplers that you would require or create them in your design libraries. Enter the coupler domain, Number of input and output pins and click the *Generate Fixed Coupler* button to generate a fixed coupler that can be used in any future design. A fixed cell coupler is created in the library that you specify. The name of the fixed cell coupler that gets created is:
`coupler_<numInputPorts>_<numOutputPorts>_[d,a]`

## Starting MATLAB

You can start MATLAB directly from ADE Explorer. If your design is using Pcell coupler blocks, then you need to create a verilog.ams file for the Pcell coupler block before netlisting. In the simulation window, choose Setup – *MATLAB/Simulink – Create Pcell Coupler File.* The Create Pcell Coupler File form appears.





© 2023 Cadence Design Systems, Inc.

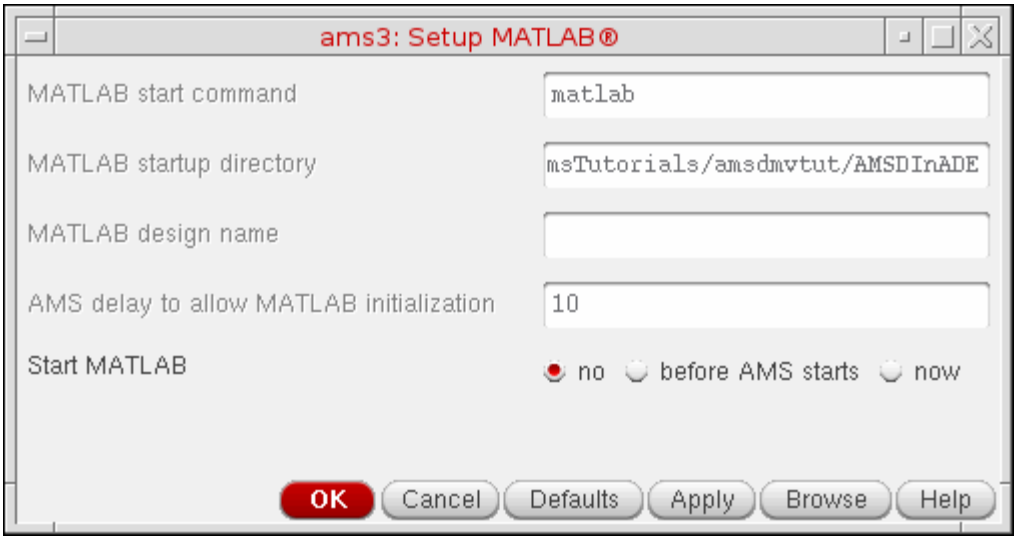
- You can either directly enter the Coupler instance or select the same from the schematic by clicking the *Select* button.

**Note:** If the block selected on the schematic is not a Pcell coupler block, the message about an incorrect selection appears in CIW.

- When you click *Generate Verilog-AMS* button, a Verilog-AMS file is created for Pcell coupler block.
- The Verilog-AMS file generated from this form is placed in the netlist directory by default. You can specify another location in the Verilog-AMS filename field.
- You can also Edit the Verilog-AMS file and save it at its original location.

After creating the verilog.vams file for the Pcell coupler, or if you are using a fixed cell coupler, choose *Setup – MATLAB/Simulink – Start.*

The Setup MATLAB form appears.




1. Enter the command for starting MATLAB using MATLAB start command field. The default command to start MATLAB is “matlab”. You can enter any other command along with the command-line argument.
2. Enter the path to the directory where matlab is launched in the MATLAB startup directory field.
3. Enter the path to the MATLAB design in the *MATLAB design name* field.
4. Enter simulation initialization timeout in AMS delay to allow MATLAB initialization. It is required to give Simulink a chance to start up before AMS. After Simulink is initialized, you can run the simulation.


© 2023 Cadence Design Systems, Inc.

**5.** Do one of the following:

| Select | IF |
|---|---|
| no | If you select the *no* option for Start MATLAB, MATLAB will not be started. This may be useful if you want to keep same form setup but do not want to run cosimulation or if you want to start MATLAB yourself, independent of ADE Explorer. |
| before AMS starts | If you want to start the MATLAB automatically, select the before *AMS starts* check box in Start MATLAB section. When you select the before AMS starts option, all the fields in the form are enabled. To run Simulink simulation automatically when AMS starts, you need to have a startup.m file that contains the sim() command in the directory, which is specified in the MATLAB startup directory field. |
| now | If you want to launch MATLAB manually, select the *now* option in *Start MATLAB*. The *Start* button appears and the AMS delay to allow MATLAB initialization is disabled. Click Start button to launch MATLAB. |


© 2023 Cadence Design Systems, Inc.

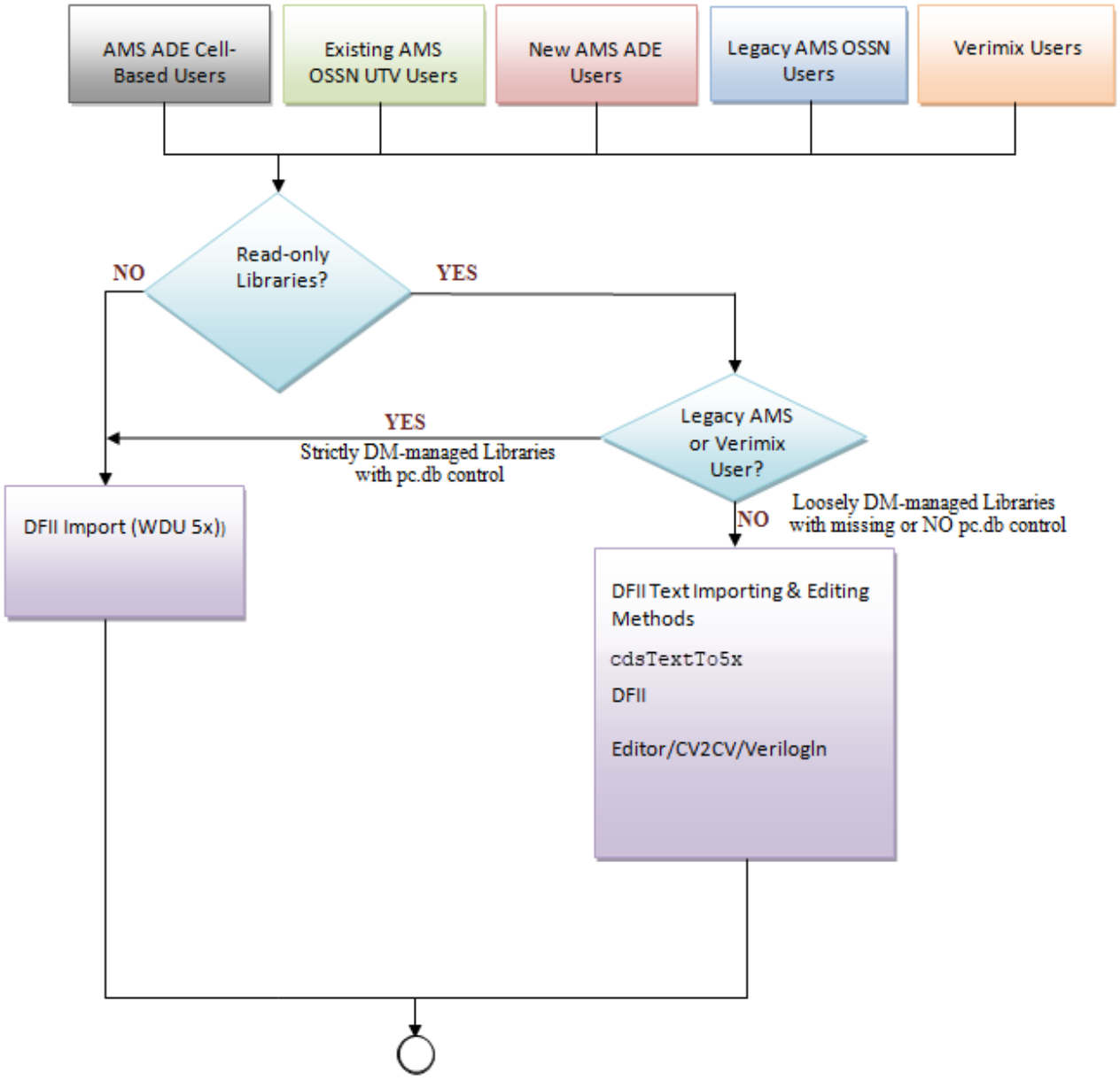
# About the AMS Unified Netlister

AMS Unified Netlister (AMS UNL) provides a comprehensive netlisting and binding framework for the Spectre AMS Designer Virtuoso use model (AVUM).

**Note:** On using AMS UNL with Xcelium 21.02 a001 or a higher release version, ensure that the environment variable `$PATH` has the Spectre 19.1 ISR15 installation path specified in it.

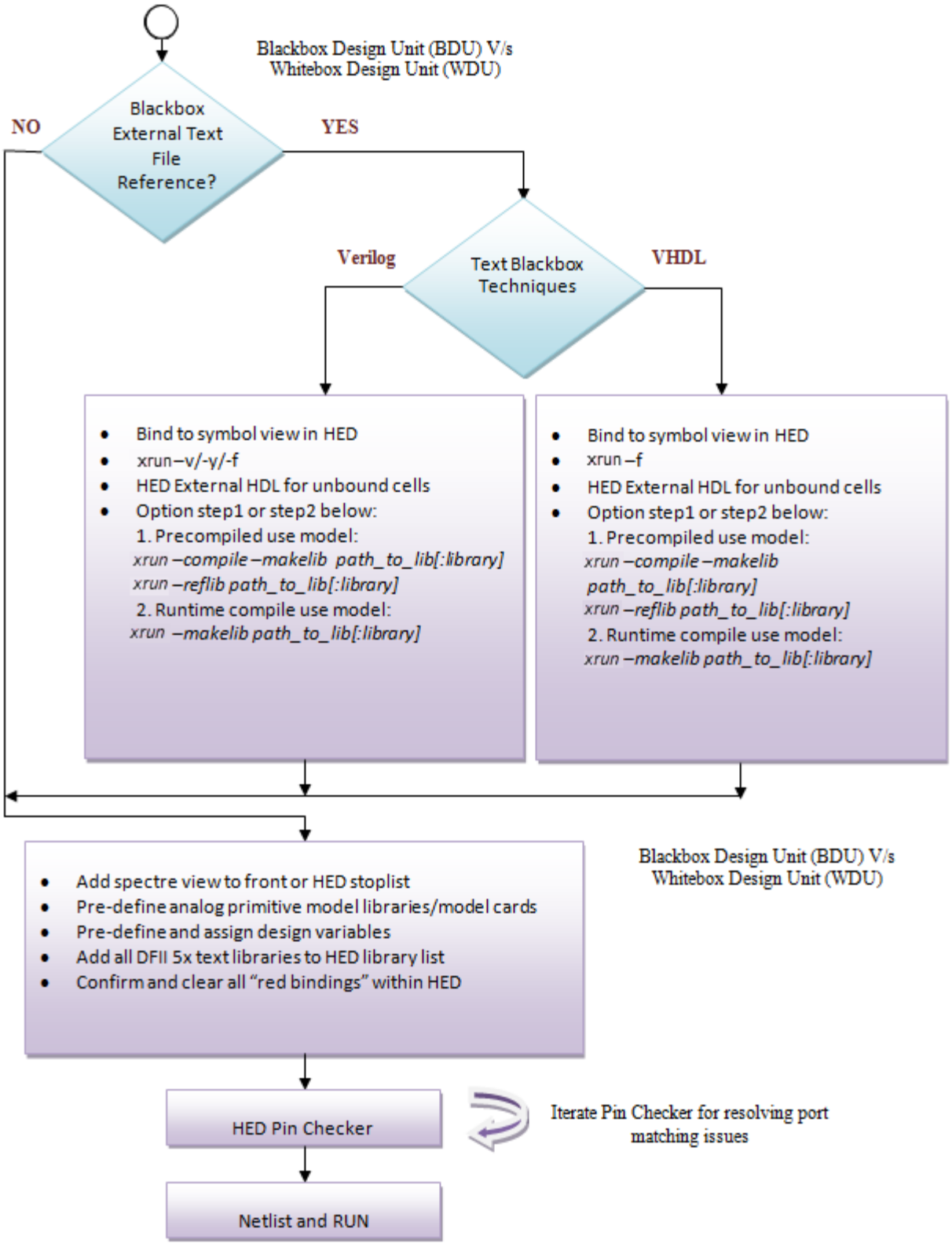

© 2023 Cadence Design Systems, Inc.

The following flowchart describes the AMS UNL flow:





© 2023 Cadence Design Systems, Inc.





© 2023 Cadence Design Systems, Inc.

## Benefits of AMS UNL

Some of the important benefits of AMS UNL are:

- It supports simulation using the new `xrun` command of the Spectre AMS Designer and Xcelium Mixed-Signal simulator. For more information about the xrun command, see the *Spectre AMS Designer and Xcelium Simulator Mixed-Signal User Guide*.
- The `runams` command provides command-line support for netlisting the design using AMS UNL and running the simulation using xrun.
- It allows better text integration and seamless migration with xrun digital-based flows using xrun external text referencing (see External Text Referencing (BDUs for Verilog/ VHDL) on page 1080) capabilities.
- It needs spectre CDFs and Spectre netlist procedures to netlist instances of analog primitives.
- It generates a single netlist comprised of all configuration cellviews that require netlisting, that is, schematic, layout, and so on. The netlist format is comprised of verilogAMS structural netlists representing the OA cellviews. The netlist can be passed to the simulator as a standalone file along with the text cellview definition file. If the simulator reports an error, you can edit the netlist file and simulate it again.
- It directly parses SystemVerilog, VHDL-AMS, and VerilogAMS text-based models.

## Things to Know When Using the AMS UNL Flow

- AMS UNL uses the Virtuoso Hierarchy Editor (HED) to configure the design units. The design units can be any of the following:
  - Netlists from DFII schematics
  - Text views imported into DFII

  In the AMS UNL flow, the following two types of configurations are possible:
  - Schematic on top
  - Text on top

  You can use the *config* view in HED to specify the top-level design unit. The rest of the design units can be configured by determining whether or not the design units are visible in HED. You can also set cellview bindings separately for individual bits of an iterated instance in HED. For more information, Refer to HED user guide.

  **Note:** If your design requires the use of two or more top-level design units, the additional


© 2023 Cadence Design Systems, Inc.

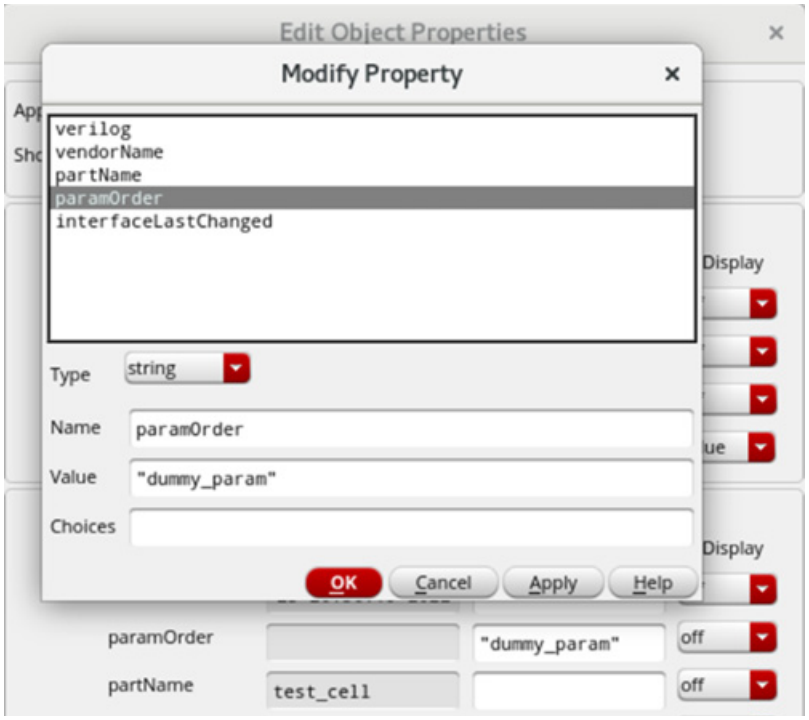
top-level module(s) can be used in parallel to your HED configuration and externally referenced by specifying the module file(s) in the *Additional files on xrun command line* field of the *AMS Options* form. You must also specify the additional top-level module(s) using the `-top <topLevelModule>` option in the *Additional Arguments* field of the *AMS Options* form.

Based on the HED configuration visibility, the design units can be classified as:

**Whitebox design units (WDUs)**: WDUs refer to the design units that are present in the DFII Library database and visible to HED. These can be schematics, layouts, or hierarchical text views. Such design units can be configured for proper cell, instance, or occurrence binding using HED.

**Blackbox design units (BDUs)**: BDUs refer to the design units that are text views where the parent-level text cellview and its hierarchy (if there is any) is invisible to HED access and netlisting. For such design units, you can configure the parent-level text cellview in HED as a stopping view. The parent-level text cellview and its hierarchy needs to be configured using xrun external text referencing specifications. The specifications for digital text referencing could be through typical xrun options, such as `-v`, `-y`, `-f`, and `-reflib`.

- If you want AMS UNL to netlist verilog hierarchical property (hierProp) parameters for stopping non-text cells, the name of those parameters must be added in the value for the paramOrder property.

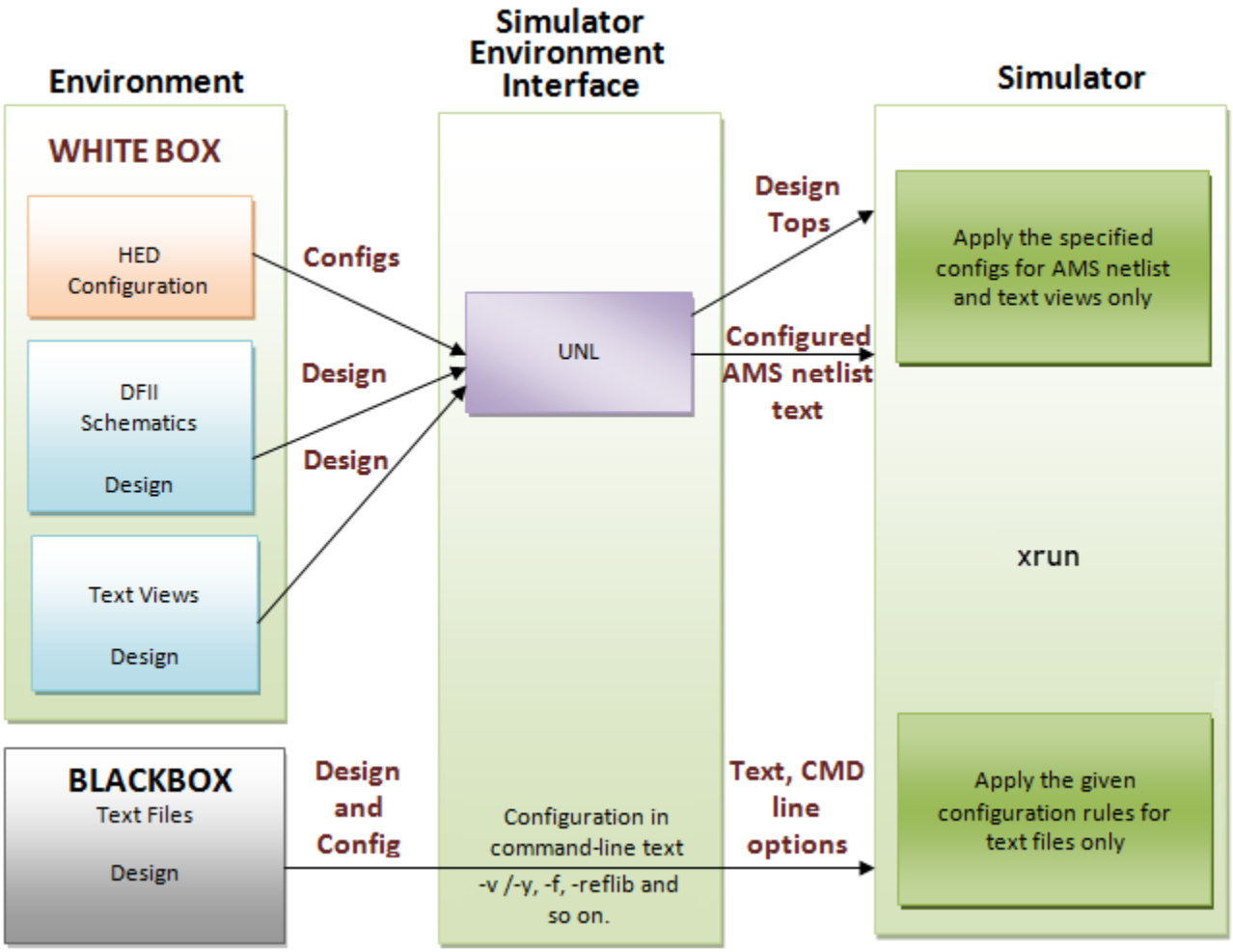




© 2023 Cadence Design Systems, Inc.

## AMS UNL Architecture

The following figure describes the AMS UNL architecture:




The Simulator Environment Interface processes the design and configuration information specified using ADE Explorer and provides the following information to the simulator (xrun):

- **Top-level design units**: These can be the top-level design units specified in the *config* view of HED.
- **Design Information**: The design information is either in the form of netlists (from schematic) and/or user-specified text files. The design units can be:
  - **Configured WDUs**: There can be two types of configured WDUs:
    - **AMS netlists**: Generated by OSSN, based on the information in HED and DFII databases. These contain the design units to be simulated by xrun.


© 2023 Cadence Design Systems, Inc.

- **Text view BDUs**: These already contain the design units to be simulated (and maybe more). They can be mixed or pure digital blocks. The configuration of the design units in the text view is visible to HED.

- **Text view BDUs**: These text views already contain the design units to be simulated. Text is referenced natively. For such text views, only the parent cellview design unit is visible to HED and bound to a stopping view (*External HDL*). These can be:
  - Digital blocks (Verilog: `-v`/`-y`, V2001, V2005, `-reflib`, `-makelib`)
  - Digital blocks (VHDL: `-f`, `-reflib`, `-makelib`)
  - Analog blocks (SPICE blocks - specified using `sourcefile` property)
  - Verilog-A blocks (specified using `ahdl_include`)
  - Precompiled libraries (pure, digital, or mixed: `-reflib`)

## AMS UNL Methodology

### *Hierarchy Editor*

The AMS UNL flow is sensitive to the HED configuration settings — *Library List*, *View List*, and *Stop List* — that are available in the *Global Bindings* section of HED. Therefore, you must ensure the following while setting these options:

### *Library List*

You must specify all the DFII 5x libraries used in the design in the *Library List* field. It is recommended to specify the libraries in the order of preferred precedence. The order can be used to influence the binding search order if there are same cells from multiple libraries.

### *View List*

You must specify all the view types used in the design in the *View List* field. It is recommended to specify the cellviews in the order of preferred precedence. The order in the list determines what a particular cell uses as its default *View To Use*. If you do not explicitly set any view for a cell, the default *View To Use* is considered as the first existing view in the list. For VHDL, add the VHDL architecture names to the HED *View List*, and bind the VHDL cellviews to the VHDL architecture names and not the entity names.


© 2023 Cadence Design Systems, Inc.

### *Stop List*

You must specify only the view names for which you wish the netlister to stop traversing the cell's hierarchy in the *Stop List* field. A hierarchical text cellview used as a stopping view in HED *Stop List* generates an error during elaboration. Stopping views must have no hierarchy or must be leaf-level. It is recommended to use *symbol* view, or a copy thereof, for text stopping views. A stopping view that has hierarchy assumes that the leaf-level hierarchical content will be found using other ways external to DFII. By default, *spectre* is considered the analog stop view and *symbol* is considered the digital stop view.

When adding or removing views from *Stop List*, configuration binding gets refreshed immediately upon an HED update. For example, when a symbol is added and a cell's *View To Use* is changed to *symbol*, the 5x hierarchy of the cell is removed from HED and is no longer visible. This assumes a blackbox condition and expects the cell's hierarchy (if there is any) to come from external sources, such as xrun external HDL specification (`-v`, `-y`, `-f`, and so on).

In addition to setting the fields in the *Global Bindings* section, ensure that all cellviews that are displayed as “red” in HED are properly resolved. You should either fix the unresolved cells in “red” manually, or bind them to an *External HDL* setting for proper blackbox resolution. This can be done by right-clicking a cell in the *Cell Bindings* section of HED and selecting the *Set Cell View - Mark as External HDL Text (AMS UNL only)* option from the context menu.

### New Binding Engine

For the AMS UNL flow, a new binding engine has been implemented. The new binding engine:

- Supports 5x formats with lib/cell/view structure.
- Supports explicit library, cell, instance, and occurrence bindings for all cell types and levels in the HED configuration.
- Supports Verilog and VHDL name spaces.
- Saves time during elaboration by quickly finding all cell/instance/occurrence bindings.
- Supports all aspects of Block-based Discipline Resolution (BDR) using the `-setdiscipline` xrun option.
- Supports HED blackbox design units (BDUs) and whitebox design units (WDUs).

  BDUs must retain external text referencing in the digital domain down to the hierarchical leaf level. That is, there is no HED 5x visibility or accessibility.


© 2023 Cadence Design Systems, Inc.

WDUs can contain any combination of OA database and/or text hierarchy in HED 5x. That is, there is full visibility and accessibility down to the leaf level with respect to HED and netlisting.

The AMS UNL flow can predefine text libraries to be either BDU or WDU, but not both.

- A text library cannot be created in WDU and used for BDU external text referencing.
- A text library precompiled with prescribed xrun methods for BDU usage cannot be used for WDU referencing.
- Any cell imported into DFII (WDU) must be in a separate library as compared to a cell that is referenced external to DFII (BDU). Do not mix 5x WDU cells with non-5x BDU cells within the same library. A library should either be WDU format or BDU format, but not both.

### Importing Text Views into Virtuoso (WDU Libraries)

The AMS UNL flow supports importing of text views into Virtuoso. You can import a text view into the Cadence 5x library/cell/view structure by using any of the following methods:

**Note:** It is recommended that you use the `cdsTextTo5x` command because it uses xrun in the background. Also, the `xmvlog -use5x` and `xmvhdl -use5x` commands have been deprecated.

- Using the xmvlog or xmvhdl `-use5x` command-line option.

  It is recommended that the text view is imported before locking the design to read-only status.

  Full 5x WDU VHDL/VHDL-AMS hierarchical support is provided to match the same support as the Verilog namespace. In addition, full hierarchical VHDL text-schematic sandwiching is supported.

- Using the `cdsTextTo5x` command. You can also use this command to generate the symbol views of the imported cellviews. For more information, see Importing Design Data by Using cdsTextTo5x.
- Choosing *Create* - *Cellview* - *From Cellview* in the schematic editor.
- Choosing *File* - *New* - *Cellview* in the command interpreter window (CIW).
- Using `Verilog In` to create Verilog structural gate-level schematics.

  `pc.db` files are not created during the `Verilog In` process because the shadow database contains only instances for which `Verilog In` is successfully able to locate the instance masters. If gate-level cells have hierarchy, you need to use the read-only


© 2023 Cadence Design Systems, Inc.

solution to allow the creation of `pc.db` files within the `tmp` directory, or manually populate the library with `pc.db` files by using the xmvlog `-use5x` command-line option. Another option is to rebind the standard cells to BDU and use external text referencing with *External HDL* binding (see External Text Referencing (BDUs for Verilog/VHDL) on page 1080).

### External Text Referencing (BDUs for Verilog/VHDL)

The AMS UNL flow supports configuration of text views as stop view or leaf, where the parent-level text view and its entire hierarchy can be made invisible to HED. For such design units, you can configure the parent-level text view in HED to be the stopping view, or use external text referencing with *External HDL* binding. The parent-level text view and its hierarchy needs to be configured using xrun specifications. AMS UNL supports the following methods:

- Traditional xrun external text referencing (`-v`/`-y`, `-f`, `-reflib`, `-makelib`)
- HED stopping cellview binding

  Bind the intended cellviews to the *symbol* view or a copy thereof. Add the *symbol* view, or the copy thereof, to the HED *Stop List* or add a stop point (see Hierarchy Editor on page 1077). You can use a new HED option (*Mark as External HDL Text (AMS UNL only)*) to enable HED binding to an external text view. Cellviews that appear as “red” in HED can be considered to be referencing an external text view. This should not be assumed and should be verified and resolved before proceeding.You can use the *Mark as External HDL Text (AMS UNL only)* option to resolve the binding for such cellviews.

#### *Text Referencing Methods*

A text view can be referenced as follows:

- Using the `-v` and `-y` arguments of xrun (references the Verilog2001 source files and Verilog 2001 Library file directories).
- Using the `-f` argument of xrun that enables you to specify the source text paths and xrun options in a text file. This is a common use model to refer to the external SystemVerilog and VHDL text source code.
- Using the `-makelib` argument of xrun that enables you to precompile the design units in the specified files into a reference library.

  ```
  -makelib path_to_library[:logical_name] source_files -endlib
  ```

- Using the precompiled libraries with the `-makelib` / `-reflib` arguments of xrun (applies to both Verilog and VHDL name spaces).


© 2023 Cadence Design Systems, Inc.

- You should precompile (recompile if the library is created using the`-use5x` option) all existing libraries with the xrun digital use model, as follows:

  ```
  xrun -compile -makelib <absolute_path_to_LIB_dir>[:logical_lib_name]
  <source><options> -endlib
  ```

  The library directory containing the precompiled source files can then be locked to a read-only state and referenced during simulation, as follows:

  ```
  -reflib <absolute_path_to_LIB_dir>[:logical_lib_name]
  ```

- You can compile all headers, packages, source text, and file-specific compile options within a single `-makelib -endlib` option.
- Multiple `-makelib` options can be specified within the `-f` file format. All order-dependent VHDL/SystemVerilog source files can be ordered appropriately in the `-f` file. The `-smartorder` option can only be used within a single `-makelib` option and cannot be used to order files across multiple `-makelib` options.

- Migrating external text from an existing digital xrun flow.
- Pre-existing digital xrun `-f` files can be used to perform `-makelib` compilations or `-reflib` precompiled library references.
- The same `-f` files can be reused in ADE Explorer as long as the library names used in the files are not the same as the names used in the Virtuoso WDU design libraries.

## Migrating to AMS UNL

If you are migrating from a Cell-Based netlisting flow to the AMS UNL flow, consider the following:

- Since HED is the master view for a cell, instance, and occurrence bindings, you must resolve all HED bindings before proceeding to the netlisting step (see Hierarchy Editor on page 1077).

  **Note:** The `CONFIG_MODELPATH` environment variable setting is obsolete in the AMS UNL flow. All HED bindings have precedence over legacy settings.

- For text modules instantiating the analog primitive models, you can use the HED binding option *Add Bind To Open (Skip Cell)* to fully unbind the analog primitives and let the default *ModelPath* library bindings take place. Ensure that there are no digital cells with the same name or with the default binding to the digital version of the cell.

  You can also use the HED binding option *Add Bind To Open (Skip Cell)* to fully unbind the analog primitives and let the custom netlist procedure operate on removing instances later from the netlist, as required. If you encounter the xmelab error `EXANCU` on instances


© 2023 Cadence Design Systems, Inc.

known to be later removed with the custom netlist procedure, use the *Add Bind To Open (Skip Cell)* option in HED for proper behavior.

- Verilog Reference File should be at leaf level.
  - Backward-compatible migration support is provided until possible deprecation -This is considered as yet another method to specify BDU leaf-level text binding. An xmelab error will occur when a Verilog Reference File has text hierarchy. It must be only leaf-level. Cell-Based flow has allowed this support with error and no guarantee for proper bindings.
- If you are using `hdl.var` files from the Cell-Based flow:
  - Avoid using `modelpath` settings
  - Remove all global settings for `-use5x` (`ncuse5x`)
  - Remove all global settings for `-ams` (`ncams`).

    It is recommended to review the ADE-added `hdl.var` file and its necessity for reuse. In most cases, AMS UNL does not require an `hdl.var` file. `NCVLOGOPTS` and `XRUNOPTS` are valid options to include in an ADE `hdl.var` file.

    The following search order is followed by AMS UNL to locate the `hdl.var` file:

    - If the path of the `hdl.var` file is specified in the *Miscellaneous* tab of the AMS Options form, it is honored by AMS UNL.
    - If the path of the `hdl.var` file is not specified in the AMS Options form, the search order defined in the `setup.loc` file is followed to locate the file. Following is the default search order defined in the `setup.loc` file:

      ```
      $CDS_INST_DIR/share/cdssetup/setup.loc
      ;   $CDS_WORKAREA
      ;   $CDS_SEARCHDIR
      ;   "~" (users home directory)
      ;   $CDS_PROJECT
      ;   $CDS_SITE
      ;   $(compute:THIS_TOOL_INST_ROOT)/share
      ```

      If the `hdl.var` file is found at a location specified by the search order, it is honored by AMS UNL and a message appears in the CIW which displays the path of the file.

      If the `hdl.var` file is not found at any location specified in the search order defined in the `setup.loc` file, no `hdl.var` file is honored.

    **Note:** If the `hdl.var` file is present in the Virtuoso launch directory but its path is neither specified in the AMS Options form nor in the search order defined in the `setup.loc` file, it is not honored.


© 2023 Cadence Design Systems, Inc.

Avoid using any global `NCVLOGOPTS` options that control specific parsers, for example, `-sv`, and `-ams`. This affects the entire xrun command line and is applied to all source files. Use the automatic xrun extension-driven parser technique. Refer to the *Spectre AMS Designer and Xcelium Simulator Mixed-Signal User Guide*for more details.

- Verilog-A cellviews retain the use of shadow for netlisting and are considered as DFII objects. These cellviews must be imported and edited in that same way.
- Ensure that the `pc.db` files do not cause HED to generate any warnings or errors. A modification to a text view stored outside Virtuoso in a read-only library/directory can generate inaccurate text representation within HED and can impact the intended UNL netlisting/binding. Use the `pc.db` read-only solution explained above to resolve the issues related to `pc.db` files.
- Ensure that the precompiled libraries that you are using are compatible with the bit version of Virtuoso. A 32-bit Virtuoso release requires 32-bit xrun precompiled libraries, whereas a 64-bit Virtuoso release requires 64-bit xrun precompiled libraries. To use both binaries of Virtuoso (32/64-bit), you need to create an xrun precompiled library using both 32-bit as well as 64-bit xrun. A library can contain both xrun binary compiled data at the same time.

## Features of AMS UNL

This section describes the following features of AMS UNL:

- Creating Text Libraries for use with the Precompiled WDU Supplemental Flow
- Supporting Parameterized Bits Using a Pcell Symbol
- Using the Verilog 2001 Configuration
- Shorting Devices
- Using the sim_stub Property
- Creating Simulation Scripts for xrun
- Reusing an Advanced Testbench
- Working with DSPF files

### Creating Text Libraries for use with the Precompiled WDU Supplemental Flow

HDL languages require file/library-specific options/bindings that must stay consistent throughout the design development cycle. The use of VHDL/SV packages need careful


© 2023 Cadence Design Systems, Inc.

attention and are better managed as precompiled objects and shared between all source code requiring them. Design cycles can be better optimized by reusing precompiled objects and sharing them with multiple design groups.

The recommended methodology is to precompile the text source/packages code ahead of time with these specific requirements and then lock-down the libraries so as not to disturb the consistent and stable "IP". At the same time, HED configurations using a visible, 5x WDU approach need to access and use these precompiled "IP" libraries in a "WDU-like" use model. This creates all possible HED 5x possibilities for library and view bindings to externally referenced text in a pseudo-BDU-like use model.

The old WDU hierarchical text flow is still the predominant and the recommended use model. This flow is supplemental to the old WDU flow such that it provides the capability to:

- Add file/library-specific options for compilation
- Precompile SV/VHDL packages
- Share precompiled objects between design groups

If a flow warrants no need for these advantages, then the old WDU 5x hierarchy should be used.

By utilizing the existing HED WDU 5x configuration settings within a Virtuoso setup, a normal WDU flow with HED 5x libraries and bindings can be achieved. This flow assumes that all WDU text cellviews visible in HED have already been imported and registered within 5x using the usual methods, that is, by using precompiled xmvlog/xmvhdl `-use5x` scripts or using traditional DFII import techniques (see Importing Text Views into Virtuoso (WDU Libraries) on page 1079).

### *Overview*

- Create a mirror-imaged precompiled, non-5x library of the same 5x DFII equivalent libraries using multiple xrun `-compile` invocations, each with its own `-makelib` options. As xrun is not 5x-aware, you need to recompile the same 5x libraries for xrun use.

  The key point here is to isolate the physical location of the 5x and non-5x versions of the same DFII libraries but retain the same logical library names.

- From the Virtuoso invoking directory, or from a central project library area accessible to all users, create a separate directory to precompile and store all source text for the duplicated, non-5x version of the same 5x DFII libraries.

  **Note:** The non-5x, precompiled library can be anywhere except in the same Virtuoso invoking directory (`cds.lib` errors may occur).


© 2023 Cadence Design Systems, Inc.

- Place a copy of the `cds.lib` file from the original Virtuoso invoking directory, except for changes to the physical library paths matching the new location, in the precompiled, non-5x directory just created. Two `cds.lib` files will exist. One is the original unmodified `cds.lib` in the Virtuoso invoking directory and the other is the `cds.lib`, with modified path locations, in the newly created precompiled, non-5x directory location.

  **Note:** The sole purpose of the modified `cds.lib` file in the non-5x directory is to aid usability in Virtuoso when referencing the precompiled non-5x `pak` files using runtime `-reflib` options. These `-reflib` options can then use the library path and logical library name as arguments when referencing the non-5x libraries using the non-5x `cds.lib` file. For now, this is a redundant step to ensure that this flow stabilizes properly. Since the ADE Explorer *REFLIB*/*MAKELIB* GUI options use both the precompiled non-5x library path AND logical libname for reflib entries, the need for this non-5x version of the `cds.lib` file will eventually not be required.

To create a precompiled library, follow these steps:

1. Create a directory with the name `precompiledLibs` in the current working directory, as follows:

   ```
   mkdir $PWD/precompiledLibs
   ```

2. Set the environment variable `XRUN_LIBS`, as follows:

   ```
   setenv XRUN_LIBS $PWD/precompiledLibs
   ```

3. Navigate to `XRUN_LIBS`, as follows:

   ```
   cd XRUN_LIBS
   ```

4. Create the precompiled, non-5x version of the DFII libraries using a script that calls out multiple xrun `-compile`, `-makelib` commands, one for each library and/or file-specific options.

5. Create and store this script at the `$XRUN_LIBS` location along with matching `cds.lib`.

6. Allow one `-makelib` statement per 5x library. Create a compile script `preCompileLibs.f` containing the following statements:

   ```
   xrun -compile -makelib <path>:<logical_lib_name> <source files> <source file specific options> -endlib
   xrun -compile -makelib <path>:<logical_lib_name> <source files> <source file specific options> -endlib
   xrun -compile -makelib <path>:<logical_lib_name> <source files> <source file specific options> -endlib
   ```

   where:

   *`path`* is the physical location of the precompiled non-5x libraries.


© 2023 Cadence Design Systems, Inc.

*logical_lib_name* is OPTIONAL and the same logical library name of the same 5x DFII WDU library designated in the Virtuoso invoking directory and called out in the path-altered copy of the precompiled, non-5x version of the `cds.lib` file.

**Example of precompiled script preCompileLibs.f**

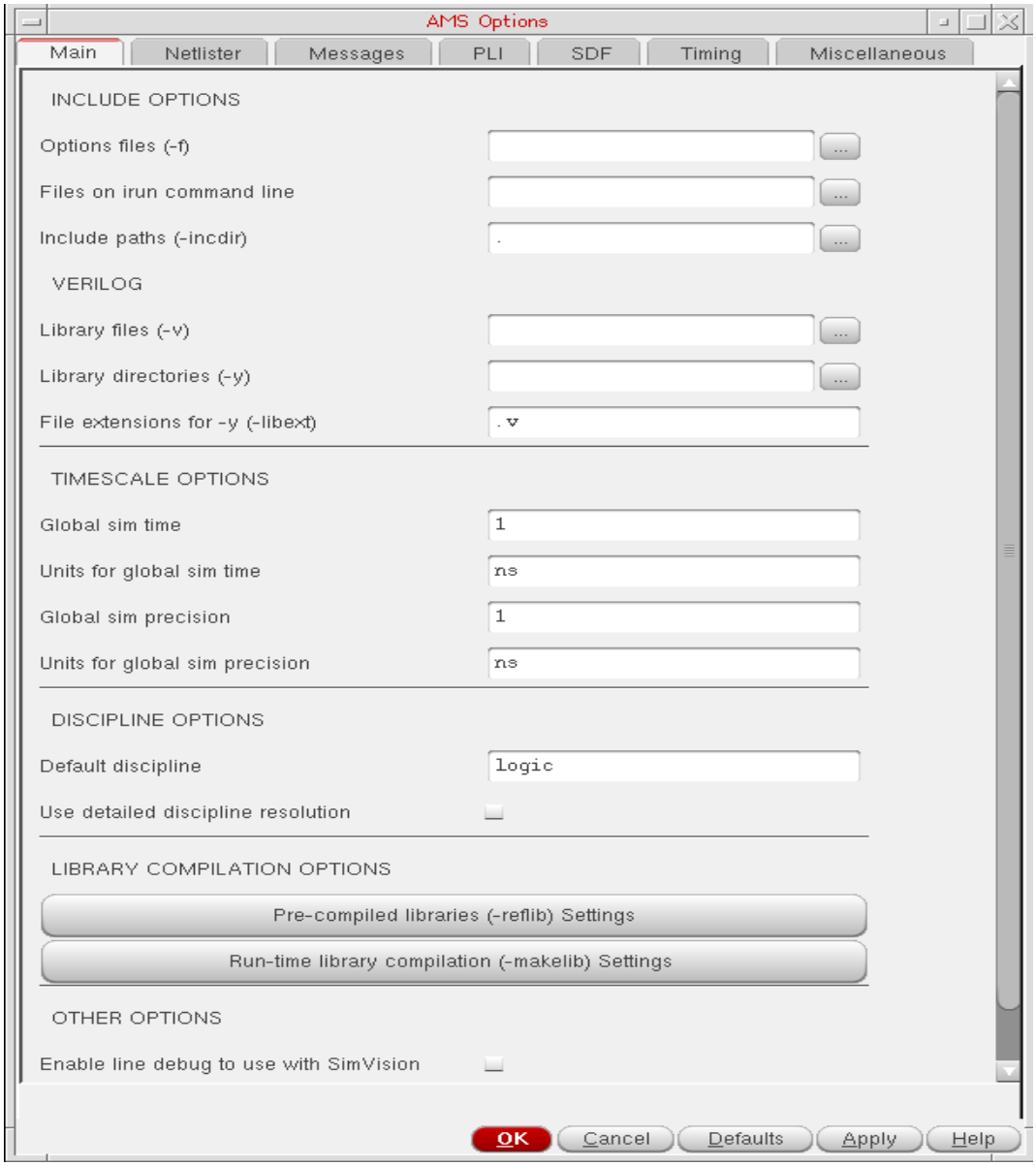
```
xrun -compile -messages
-makelib ./DigLib $PROJ_LIBS/source/DIG/*.vhd -v93 -relax -endlib
xrun -compile -messages
-makelib ./AMSLib $PROJ_LIBS/source/AMS/*.vhms $PROJ_LIBS/source/AMS/*.vams -
endlib
xrun -compile -messages
-makelib ./stdLib $PROJ_LIBS/source/STD/*.vhd -v93 -endlib
xrun -compile -messages
-makelib ./RTLLib $PROJ_LIBS/source/RTL/*.vhd -v93 -endlib
xrun -compile -messages
-makelib ./testLib $PROJ_LIBS/source/test/*.vhd -v93 -relax -endlib
xrun -compile -messages
-makelib ./HDLLib $PROJ_LIBS/source/HDL/*.sv -endlib
…
```

Where `$XRUN_LIBS` is predefined as `$PWD/precompiledLibs`.

7. Run the `xrun -f preCompileLibs.f` script from the `$XRUN_LIBS` location and then lock down the precompiled library directory as reusable "IP".


© 2023 Cadence Design Systems, Inc.

8. Add the precompiled libraries in the *Reference Library Setup* form by clicking the *Pre-compiled libraries (-reflib) Settings* button in the *AMS Options* form.




9. Remove all other previously referenced xrun `-f` files that are now referred to with the same reflib library paths/name using the *REFLIB*/*MAKELIB* options.

An additional feature has been implemented for automatic and seamless VHDL/SV package collection and compilation for the simulation phase of the UNL flow. The advantage of this feature is the ability to carry through to the xrun simulation stage any precompiled VHDL/SV packages that were found used and precompiled in the design libraries. This removes the


© 2023 Cadence Design Systems, Inc.

need to search and collect all packages required for simulation. The tool does this automatically and efficiently.

The precompiled WDU libraries can now be referenced as reusable "IP" with open access to all AMS design teams working on a project. This allows reusing the libraries as shared and locked-down "IP". Digital design teams can manage these external reference libraries at their own discretion. Changes made to the external text source will be realized by the analog teams running Virtuoso by using the *Pre-compiled libraries (-reflib) Settings* option in ADE Explorer. This allows for seamless integration between the GUI and command-line use models. Design cycle productivity can be greatly enhanced when using precompiled, external text with this methodology. Time is saved during netlisting and compilation iterations by using external reference libraries.

### Supporting Parameterized Bits Using a Pcell Symbol

In digital designs, parameterized port sizes are often used as a design reuse practice. However, in case of AMS Designer, the implementation of symbols does not support this practice. Therefore, in order to use parameterized bits, you need to create a Parameterized Cell (Pcell) symbol for the module instead of a symbol. A Pcell is a programmable cell defined in a way that each instance of the cell can be customized. And a schematic Pcell allows the connectivity of the cell to change according to parameters that are specified for each instance of the cell.

Consider the following module containing an output pin named `out`:

```
module xyz (out,in);
parameter integer bits=26;
output[0:bits] out;
input in;
logic out;
logic in;
assign out = ~in;
Endmodule
```

The module `xyz` can be reused in several designs with each having a different number of bits. In this case, bits is an integer parameter that controls the variable bus width of the output pin, out. When a Pcell symbol is created for this module, it contains a pin named `out<0:26>` and changing the value of the instance parameter, bits, changes the bus width of the output pin, out, for that instance of the module. When you change the value of bits, the port name for the Pcell symbol also changes accordingly. Therefore, there can be multiple instances of the same module with varying widths of the bus.

You can either create a new Pcell symbol or convert an existing symbol into a Pcell symbol and then use the Pcell symbol in your design, which can be netlisted in AMS UNL.


© 2023 Cadence Design Systems, Inc.

To know how to create a Pcell, see Creating SKILL Parameterized Cells in the *Virtuoso Parameterized Cell Reference*.

To convert an existing symbol into a Pcell symbol, perform the following steps:

1. Open the symbol in the Virtuoso Schematic Editor and enter the following code in the CIW:

   ```
   dbWriteSkill(geGetWindowCellView() "mysymbol.il" "w" "4.4")
   ```

   A file named `mysymbol.il` is created. The code in this file contains all the correct coordinates, matching labels, and the net information.

2. In the code generated in the `mysymbol.il` file, the pin name is a fixed pin name as shown below:

   ```
   dbD_0xc6384f4 = dbCreateNet(dbD_0xc62882c "out<26:0>" nil)
   ```

   Parameterize the pin names in the code as shown below:

   ```
   dbD_0xc6384f4 = dbCreateNet(pcCellView sprintf(op "out<%d:0>"
   bits) nil)
   ```

After creating the Pcell, perform the following steps:

1. Load the file `mysymbol.il`, which contains the code of the Pcell, in CIW to generate the Pcell symbol.
2. You can create a new verilogAMS cellview for the Pcell symbol or use an existing verilogAMS cellview of the Pcell symbol.
3. Netlist the design.

### Using the Verilog 2001 Configuration

UNL supports Verilog 2001 (V2001) configuration, which allows you to configure the contents of a design. The V2001 configuration provides the flexibility of configuring every instance in Verilog language.

**Note:** It is recommended that you compile the Verilog 2001 config using the `-compcnfg` option as the `-libmap` option does not support the use of `-v`/`-y` xrun options.

Perform the following steps to enable Verilog 2001 in UNL:

1. Choose *Simulation – Options – AMS Simulator*.

   The AMS Options form appears.


© 2023 Cadence Design Systems, Inc.

**2.** Perform the following steps:

- To set the Verilog config block and UNL top as the same-top, wherein the Verilog config top and UNL top have the same top levels, perform the following steps:
  - Specify the following xrun options in the *Additional Arguments* field on the Main tab:
    `myconfig.v -compcnfg -top cfg1`
  - Add the code for `cfg1` in the Verilog configuration file, `lib.map` or `myconfig.v`, as shown below:

```
config cfg1;
    design zambezi45_sim.tb_dut;
    //    Set default library search order
    default liblist  behav_lib analog_lib gate_lib;
    //    Select gate-level implementation for DUT0.m0 instance of MUX
    instance tb_dut.DUT0.m0 liblist analog_lib;
endconfig
```

  **Note:** Ensure that the parameter `design` in the Verilog configuration file, such as `zambezi45_sim.tb_dut` in the code for `cfg1` given above, is the same as the top-level design unit specified in the *config* view of HED.

- To set the Verilog config block as the sub-top, wherein the Verilog config top and UNL top are the same, but Verilog config uses a sub-config which does not match the top , perform the following steps:

  - Specify the following xrun options in the *Additional Arguments* field on the *Main* tab:
    `myconfig.v -compcnfg -top cfg1`
  - Add the code for `cfg1` in the Verilog configuration file, `lib.map` or `myconfig.v`, as shown below:

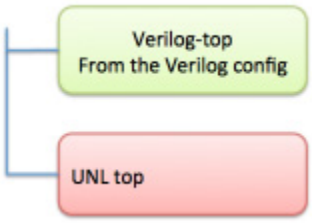
```
config cfg1;
    design zambezi45_sim.tb_dut;
    //    Set default library search order
    default liblist  behav_lib analog_lib gate_lib;
    //    Select gate-level implementation for DUT0.m0 instance of MUX
    instance tb_dut.DUT0.m0 liblist analog_lib;
endconfig
```

  **Note:** Ensure that the parameter `design` in the master Verilog configuration file, such as `zambezi45_sim.tb_dut` in the code for `cfg1` given above, is the same


© 2023 Cadence Design Systems, Inc.

as the top-level design unit specified in the *config* view of HED.

- To set the Verilog config block and UNL top as the parallel-top, wherein the Verilog config top and UNL top are at the same level of design, perform the following steps:
  - Specify the following xrun options in the *Additional Arguments* field on the Main tab:
    `myconfig.v -compcnfg -top cfg1`
  - Add the code for `cfg1` in the Verilog configuration file, `lib.map` or `myconfig.v`. In this case, the parameter `design` in the Verilog configuration file does not have to be the same as the top-level design unit specified in the *config* view of HED.




**Note:** The additional statement, `-top cfg1` is mandatory for using the Verilog 2001 config with UNL.

## Exporting IP Blocks in AMS UNL

With AMS UNL, you can export a DUT IP block (database), which includes the netlist, text files, and configuration, and then reuse it later in different testbenches running simulations with the Spectre AMS simulator. It helps simulate the design without having the need to provide absolute paths and then configure the complete design on the System on Chip (SoC) level. By default, the exported IP is saved in your current working directory, however, you can also specify the location where you want to save the IP database.

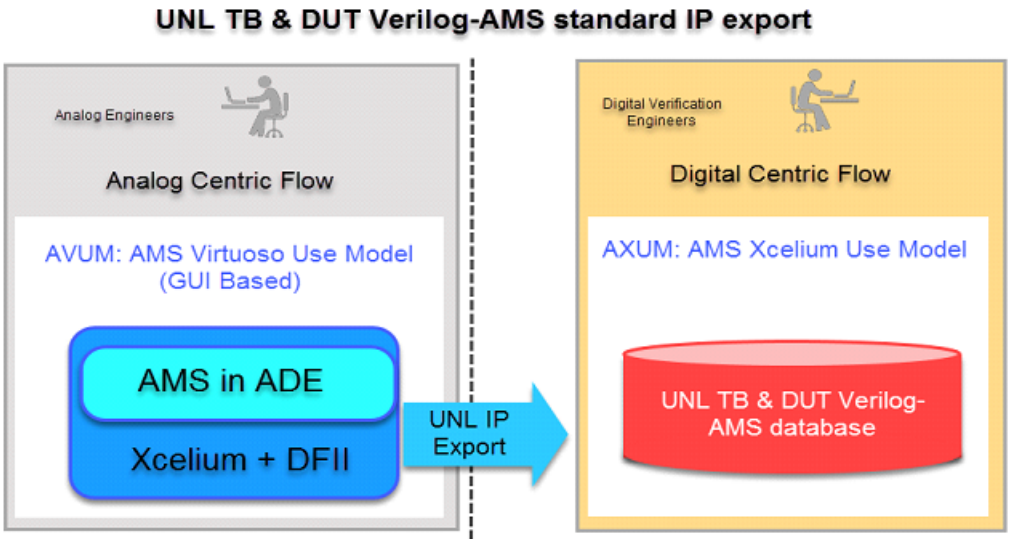
**Note:** The exported IP database can be reused after updating the extracted setup files that contain the definitions for the LINUX file paths.

*Important*

This feature is available only in Xcelium 17.04 or higher versions.


© 2023 Cadence Design Systems, Inc.

The following figure shows the architecture for IP export in analog and digital design flows:

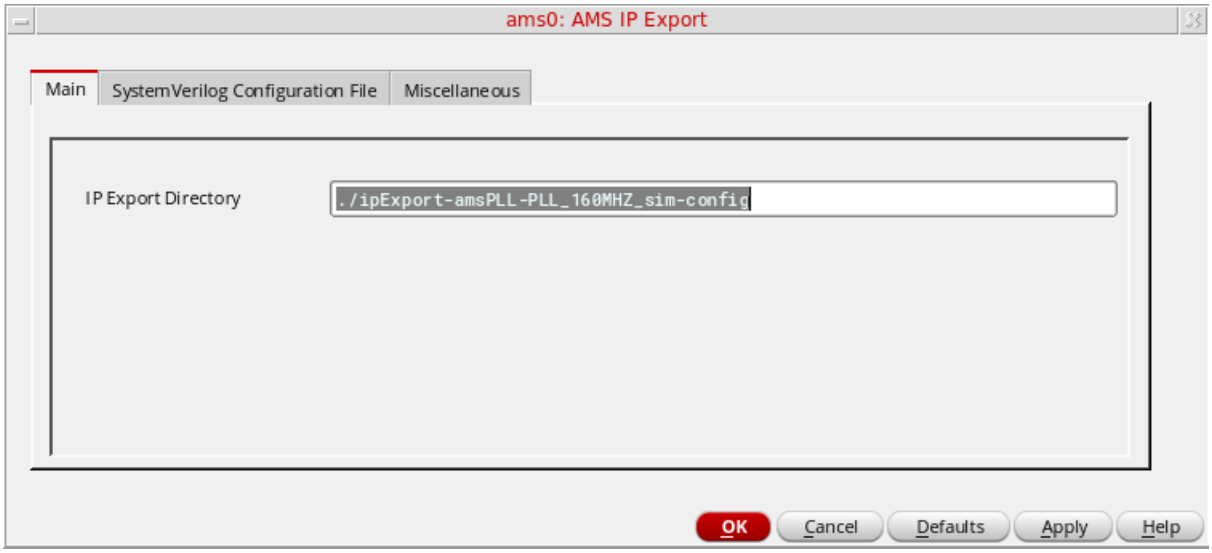



To export the netlist and other binding information (referred as IP),

➡ Choose *Tools – Export – Export as IP block*.

The AMS IP Export form appears.

**Note:** The *Export as IP block* menu option becomes available only when you have the netlist available for the current design. If not, you need to run a simulation to obtain the netlist.

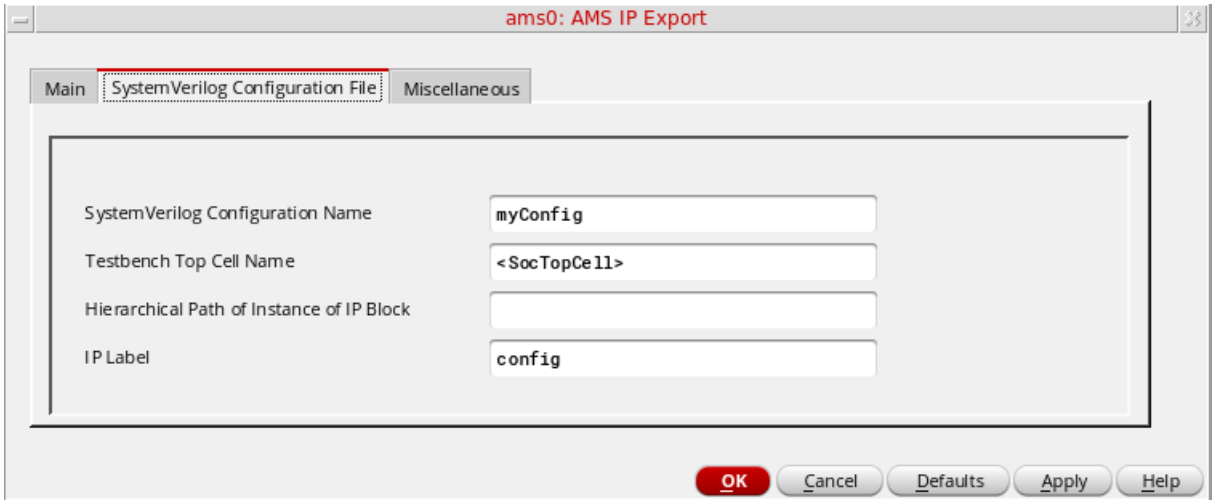




© 2023 Cadence Design Systems, Inc.

This form includes the following three tabs:

- *Main*:
  - *IP Export Directory*—Directory path at which the IP database is to be exported. By default, it is exported in the current working directory.
- *SystemVerilog Configuration File:*

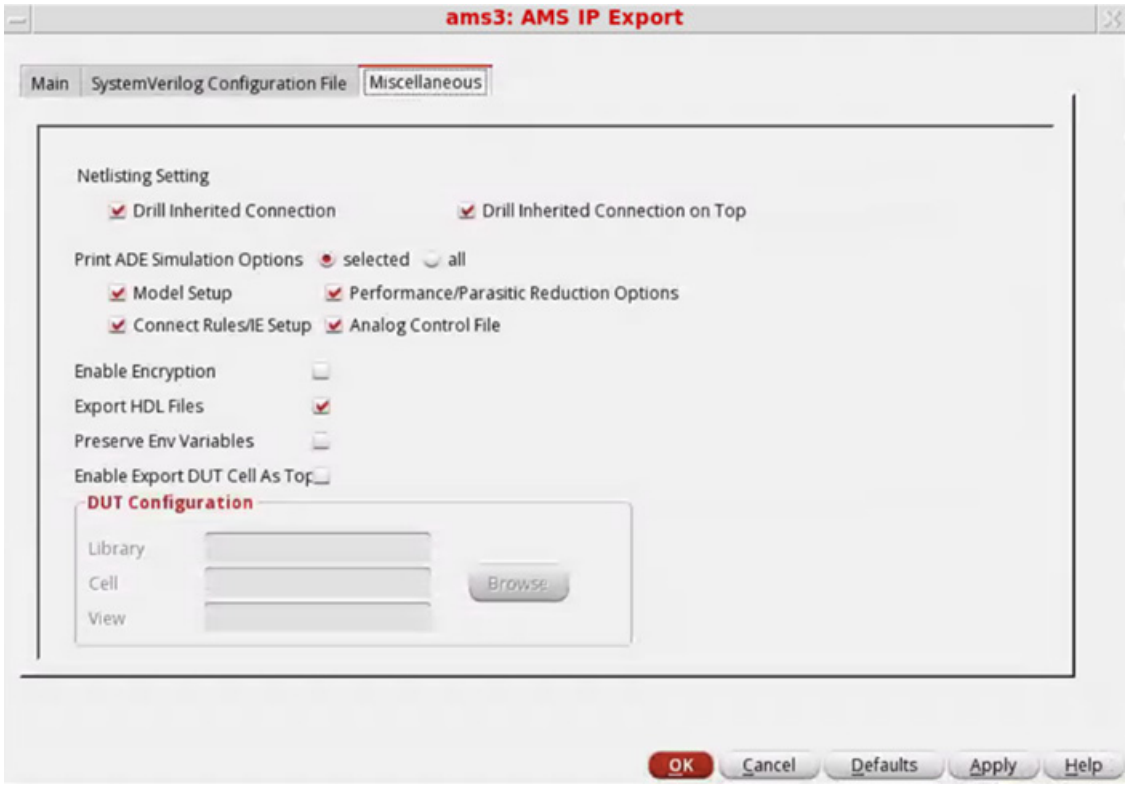



  - *System Verilog Configuration Name*—Specify the configuration name to be used for the Verilog 2001 configuration file generation.
  - *Testbench Top Cell Name*—Name of the testbench where the exported IP is to be instantiated.
  - *Hierarchical Path of Instance of IP Block*—Hierarchical path of the instance in the testbench where the exported IP is to be instantiated.
  - *IP Label*—Specify the label name to be used in the exported directory name. This label is a string value used to differentiate the AMSD bindings setup when you have multiple netlists exported as IPs. By default, the configuration view name is used as the IP label.

**Note:** For example, the name of the exported IP directory is displayed as `ipExport-amsPLL-PLL_160MHZ_sim-config`, where `ipExport` is the predefined label, `amsPLL` is the library, `PLL_160MHZ_sim` is the cell name, and `config` is the IP label.


© 2023 Cadence Design Systems, Inc.

- *Miscellaneous*:




- *Netlisting Setting*—Select the following check boxes to specify which inherited connections are to be printed in the `.vams` file.
  - *Drill Inherited Connection*
  - *Drill Inherited Connection on Top*

  **Note:** These check boxes are selected by default.
- *Print ADE Simulation Options*—Specify which ADE simulation options are to be printed during IP export. Select *all* to choose all the options and *selected* to choose all or some of the following options:
  - *Model Setup*
  - *Performance Options*
  - *Connect Rules/IE Setup*
  - *Analog Control File*
- *Enable Encryption*— Select this check box to encrypt the exported IP directory to protect it from unauthorized access.


© 2023 Cadence Design Systems, Inc.

**Note:** When *Enable Encryption* is selected, you will not be able to read the source file. The respective warning messages are displayed if any issues are found during the compilation, elaboration, and simulation operations in the IP database. The information about encrypted filenames and source location is not shared in the messages.

- *Export HDL Files*— Select this check box to copy the HDL files to the specified IP Export path.
- *Preserve Env Variables*— Select this check box to save all the environment variables set while working on the AMS IP Export flow. These variables are saved in a file named `exported_env_vars.sh` in the IP export directory.

  If required, you can also modify the values of the variables saved in this file and source it before running a simulation to set all the environment variables as specified during the IP export.
- *Enable Export DUT Cell As Top*— Select this check box to export the IP block from the specified DUT configuration.

  In the *DUT Configuration* group box, specify the details in the following fields.
  - *Library*— specify the DUT library name.
  - *Cell*— specify the DUT cell name.
  - *View*— specify the DUT view name.

To know more, refer to the AMS IP Export Reuse Flow rapid adoption kit on Cadence Online Support.

To know about the `runams` options used to export the netlist and other binding information, see IP Export Options.

To view the complete list of `runams` command-line options, see Using the runams Command.

### Shorting Devices

You can use Hierarchy Editor (HED) or the `shortInstanceList` environment variable to short devices when using the AMS Unified Netlister.

See the next topic for information on how to short devices using HED.


© 2023 Cadence Design Systems, Inc.

### *Shorting Devices Using Hierarchy Editor*

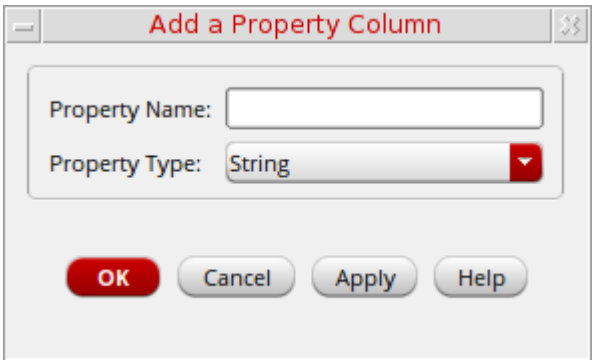
You can set the property `unlconn` in Hierarchy Editor (HED) to short devices. This property can be set on the cell, instance, or occurrence level in HED.

**Note:** Currently this property is supported only for devices with two terminals.

Open the config corresponding to your design in HED and perform the following steps:

1. Choose *View – Properties* to view the properties applied on each instance.
2. Choose *Edit – Add Property Column* to add a new property.

   The Add a Property Column form appears.



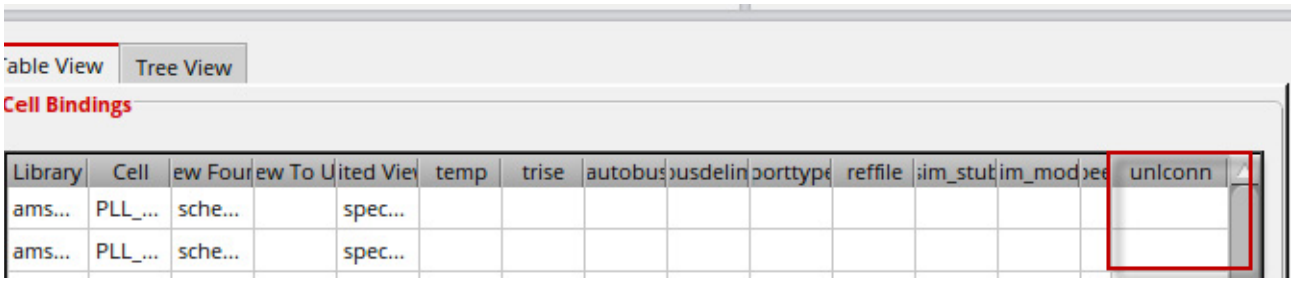

Perform the following steps:

**a.** Specify the name of the property as `unlconn`.

**b.** Ensure that `String` is selected in the *Property Type* drop down list box.

**c.** Click *OK* or *Apply*.

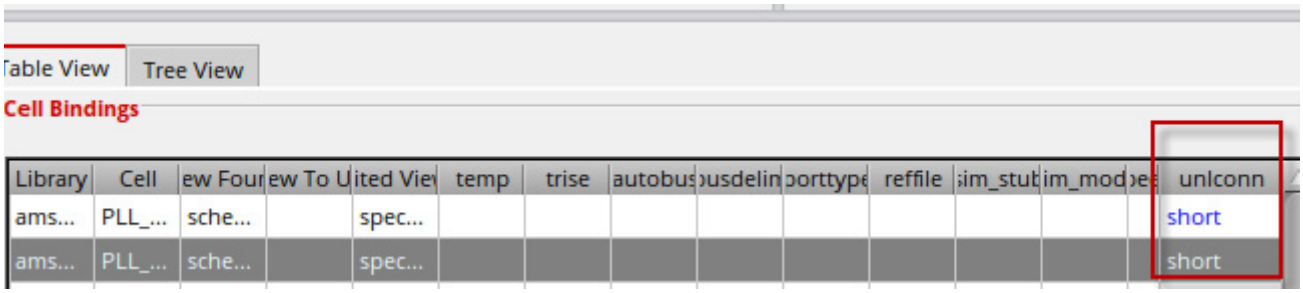
A new property column, *unlconn,* is added in HED as shown below:





© 2023 Cadence Design Systems, Inc.

3. Specify the value `short` in the *unlconn* property column next to the cell or instance you want to short, as shown below:




When you netlist the design, the cell or instance, for which you have set the `unlconn` property, are shorted in the generated netlist.

### Other Important Points to Note

- This property is supported only for the AMS UNL netlister.
- This property must be set only on a stopping device. If this property is set on a non-stopping device, an error message is displayed in the CIW.
- Since HED properties are stored in the HED file `prop.cfg`, you need to create a new config for designs in which devices are to be shorted.
- This property does not short a device globally. However, you can set this property on all the devices of a particular master in the *Table View* tab. This needs to done for each config separately.

### Using the sim_stub Property

The `sim_stub` property removes a bound schematic, Verilog-A, Verilog-AMS, Spectre, or SPICE block from the configuration and replaces the schematic or block with a stub module that is empty except the interfaces and discipline declarations.

By eliminating portions of your design, this property helps you identify the blocks that slow down the simulation. However, your design may not simulate correctly if you eliminate the circuitry for essential functions, such as signal generation.

For more information about this property, see sim_stub Property in *Spectre AMS Designer and Xcelium Simulator Mixed-Signal User Guide*.

Stubs are of two kinds:


© 2023 Cadence Design Systems, Inc.

- Digital — Such as Verilog-AMS modules. You can specify the value of the `sim_stub` property for such modules in the *`lib.cell:view`* format.

  For example, consider a cell, `vco`, which is a Verilog-AMS module with a cell-based binding configuration. The following steps show how to stub out `vco` using HED:

  **a.** Choose *View – Properties*. The *sim_stub* property column is displayed in both table view and tree view.

  **b.** In the table view, enter `amspll.vco:verilogams` in the *sim_stub* property column for `vco`. Alternatively, to automatically set the value, right-click in the *sim_stub* property column for `vco` and select *Set Cell Property*.

- Analog — Such as Spectre or SPICE blocks. You can set the `sim_stub` property for a cell, instance, or an occurrence binding configuration in the `{-cell` *`cell`*`} -src` *`sourcefile`* format.

  For example, consider an instance, `I0`, whose SPICE master subcircuit is `phaseDetPath` in an instance-based binding configuration. The following steps show how to stub out `I0` using HED:

  **a.** Choose *View – Properties*. The *sim_stub* property column is displayed in both table view and tree view.

  **b.** In the tree view, enter `-cell phaseDetPath -src $CDIR/phaseDetPath.scs` in the *sim_stub* property column for the `phaseDetPath`.

**Note:** You can use the `-libverbose` xrun option to add the design unit resolving information in the `xrun.log` file.

### Creating Simulation Scripts for xrun

Use the *Simulation scripts for xrun* option to export the simulation files from the netlist directory. This option copies all the files required for running the simulation, and generates the `xrunScript` file in the export directory. You can run the `xrunScript` file to run the AMS simulation from the command-line.

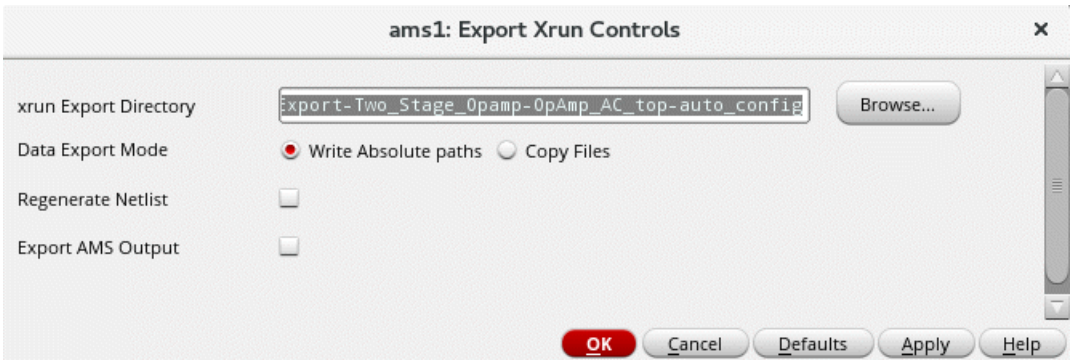
**Note:** The *Simulation scripts for xrun* option copies all the files from the netlist directory except the definition, model, and stimulus files.

To export the simulation files,

1. Choose *Tools – Export – Simulation scripts for xrun*.


© 2023 Cadence Design Systems, Inc.

The *Export xrun Controls* form appears.




2. In the *xrun Export Directory* field, specify the directory where the simulation files are to be exported.
3. In the *Data Export Mode* field, select one of the following options:
   - *Write Absolute paths*—Select this option to use the absolute paths in the `textInputs` file for the text view.
   - *Choose Copy Files*—Select this option to use relative paths in the `textInputs` file. This option copies all the files, including the sub-directories, referred in the textInputs file from the cellview directory to the export directory.

   **Note:** The `textInputs` file contains the text files for text views in the design.
4. Regenerate the netlist file and other simulation input files before exporting by selecting the *Regenerate Netlist* check box.
5. (Optional) Select the *Export AMS Output* check box to create an OCEAN file `amsOutput.ocn` in the specified directory that saves all the output expressions along with the amap file. This amap file is saved in the *`xrunExportDirectory`*`/netlist` directory.
6. Click OK.

### Reusing an Advanced Testbench

The Advanced Testbench Reuse mode lets you reuse an existing verification setup and a top-level module to reduce the verification effort. With this mode, you can reuse a digital super top above the AMSD HED configuration that you are simulating in ADE.

For example, assume that you have created an AMSD ADE testbench with an HED configuration. In parallel, the digital team has created a UVM testbench above the design under test (DUT) simulated in the AMSD ADE flow. Now, if you want to reuse the Digital on


© 2023 Cadence Design Systems, Inc.

Top (DoT) setup from the digital team's setup, the Advanced Testbench Reuse form lets you do so in an efficient manner.

See Benefits of Reusing an Advanced Testbench for more information.

There are two modes of reusing an advanced testbench:

- Full Mode—Automatically creates a SystemVerilog configuration file using the options you set to configure the DoT partitioning configuration in the Advanced Testbench Reuse form. This mode automatically instantiates the AMSD HED configuration and allows to configure HDL instances outside the DUT by using the same SystemVerilog configuration.

  You can use this mode if you do not have a SystemVerilog configuration for the top-level testbench in place.

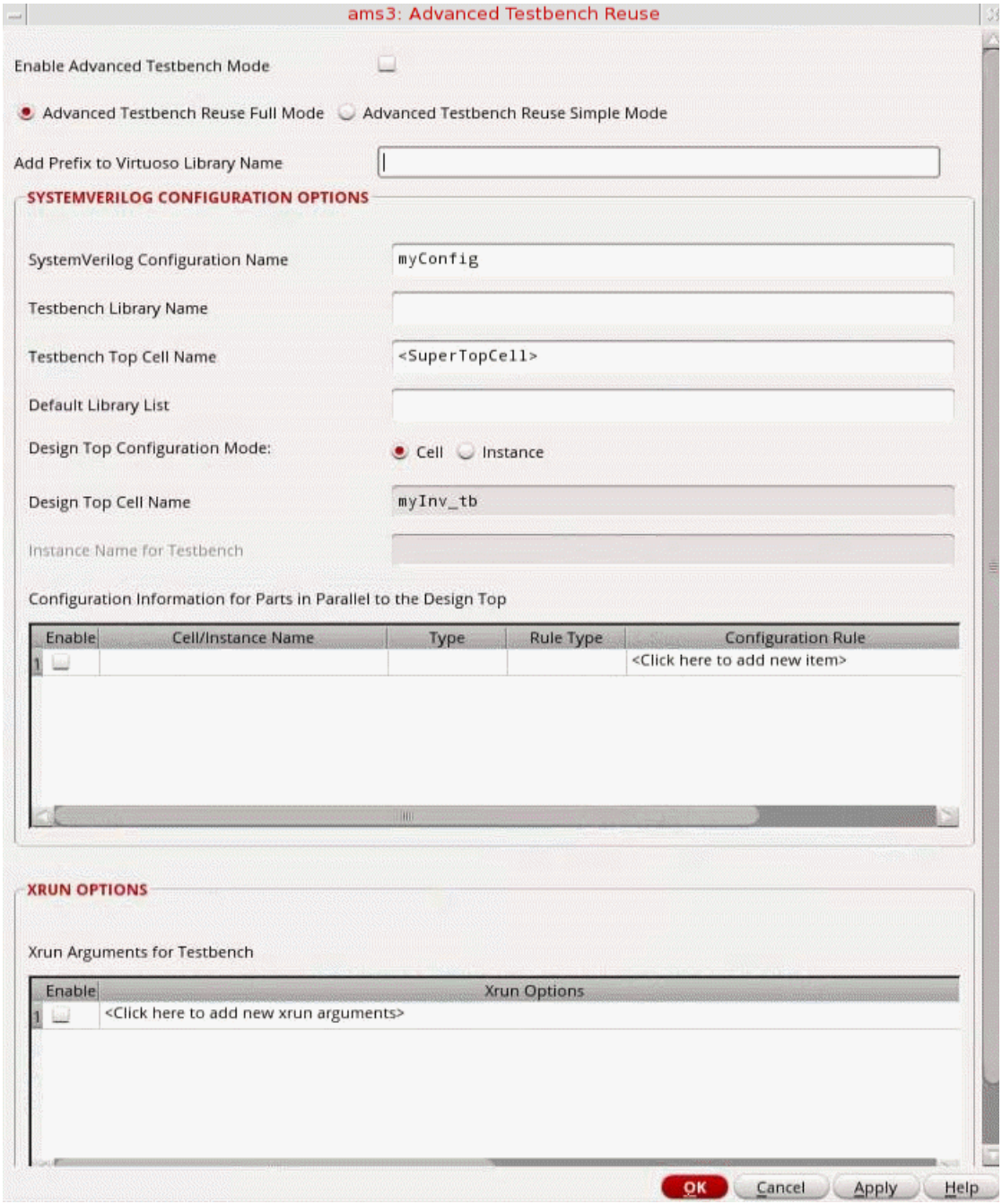
- Simple Mode—Uses an existing DoT SystemVerilog configuration file, and reuses all the options, files, and setup coming from the digital flow. The configuration file only needs to be extended in order to define the desired instantiation of the AMSD HED configuration.

  You can use this mode if you have the DoT SystemVerilog configuration to place above the AMSD HED configuration.

Perform the following steps to enable the Advanced Testbench Reuse mode and set the required options:


© 2023 Cadence Design Systems, Inc.

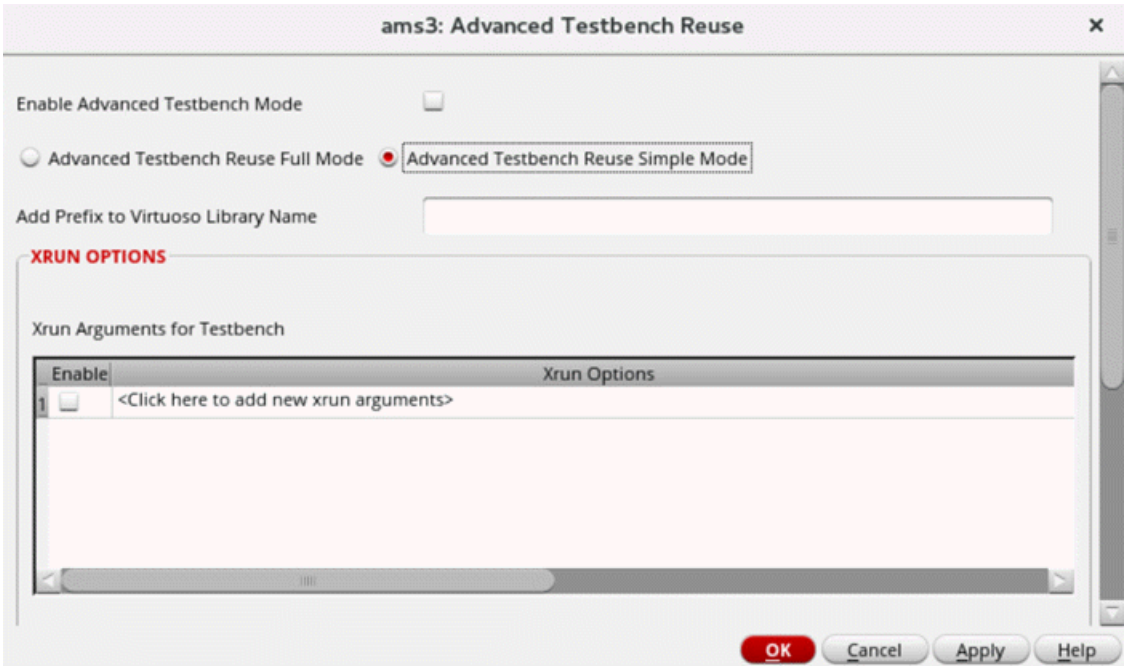
1. Choose *Tools – Advanced Testbench Reuse*. The Advanced Testbench Reuse form appears.




2. To enable this mode, select the *Enable Advanced Testbench Mode* check box.
3. Select one of the following modes:


© 2023 Cadence Design Systems, Inc.

- *Advanced Testbench Reuse Full Mode*—Enables the Full Mode for reusing an advanced testbench. This mode is selected by default.
- *Advanced Testbench Reuse Simple Mode*—Enables the Simple Mode for reusing an advanced testbench and the following options are displayed.




In this mode, a SystemVerilog configuration file is not generated. If you have an existing SystemVerilog configuration for your testbench, pass it to xrun using the `-f` option in the *xrun Arguments for Testbench* table. Also, define the top cell name using the `-top` option in the *xrun Arguments for Testbench* table.

4. (Optional) In the *Add Prefix to Virtuoso Library Name* field, specify a prefix for the Virtuoso library name.
5. In the *SYSTEMVERILOG CONFIGURATION OPTIONS* section, specify the values for the following options:

   **Note:** This section appears only when *Advanced Testbench Reuse Full Mode* is selected.

   - *SystemVerilog Configuration Name*—Specify a name for the SystemVerilog configuration file to be generated.
   - *Testbench Library Name*—Specify the name of the library from which the top-level source description is to be used for the SystemVerilog configuration file.


© 2023 Cadence Design Systems, Inc.

- *Testbench Top Cell Name*—Specify the name of the top-level cell from which the source description is to be used for the SystemVerilog configuration file.
- *Default Library List*—Specify the libraries to be searched for the source descriptions to be added to the SystemVerilog file. These libraries are searched in the order in which they are specified. By default, all the subinstances of the top-level cell in the specified library are searched, which means that the source descriptions in all the `.sv` files or `.v` files mapped to these subinstances are used.
- *Design Top Configuration Mode*—This can be set to one of the following:
  - *Cell*—Select this option if you want all the subinstances in the specified top-level cell to be used for the instance binding of the DUT.
  - *Instance*—Select this option if you want only specific instances to be used for the instance binding of the DUT.
- *Design Top Cell Name*—Displays the name of the design top cell name. This field cannot be edited.
- *Instance Name for Testbench*—Specify the names of the instances to be used for the instance binding of the DUT. Separate the names by a space character.

  **Note:** This field is disabled by default. It is enabled only when the *Instance* option is selected in the *Design Top Configuration Mode* field.
- *Configuration Information for Parts in Parallel to the Design Top*—In this table, you can specify additional instance or cell bindings for unconfigured design parts in parallel to the DUT.

  If you want the binding information in a row to be added to the generated configuration file, click the *Enable* check box in that row.

  To move a row up or down, or to delete a row, right-click that row and choose an appropriate option.

6. In the XRUN OPTIONS section, specify the additional xrun arguments in the *xrun Options* column of the *xrun Arguments for Testbench* table.

   If you want the xrun arguments specified in a row to be added to the `xrunArgs` file, click the *Enable* check box in that row.

   To move a row up or down, or to delete a row, right-click that row and choose an appropriate option.

7. Click *OK* or *Apply* to save the setup.

   When you netlist the design or run the simulation in this mode, the specified xrun options are added to the `xrunArgs` file. The `xrunArgs` file along with the generated or specified


© 2023 Cadence Design Systems, Inc.

SystemVerilog configuration file is then passed to xrun for the top-level verification of complete Systems on Chip (SoCs).

### *Benefits of Reusing an Advanced Testbench*

- Complex use cases need to be created only once. This reduces the project overhead.
- The need to import a top-level Verilog, SystemVerilog, or VHDL testbench in Virtuoso is eliminated. No additional scripts are required for importing testbenches. This helps in keeping the OA versions synchronized with the DoT external text files.
- An existing digital verification setup (for example, `.f` files) can be used as is.
- SystemVerilog or Universal Verification Methodology (UVM) testbenches can be reused without a manual setup.
- Inconsistencies in the DUT are detected.
- The need to recreate complex stimuli in Virtuoso is eliminated because you can reuse a DoT stimuli from an existing setup.

### *Examples of Configurations Used in Simple Mode*

Following are a few examples of custom SystemVerilog configurations that can be used in the Simple Mode:

- For instance binding, that is, when only one instance of module `block_in_virtuoso` uses the HED configuration:

```
config myconfig_verilog_top;
        // design top level (DoT)
        design testLib.verilog_top_block_external_as_text;
       // instantiation of the HED configuration for module "block_in_virtuoso"
        instance verilog_top_block_external_as_text.INST_TOP_IN_VIRTUOSO use
cds_amsconfiglib.block_in_virtuoso:amsconfig;
       // instances of module "verilog_block_external_as_text" which are part
of the DoT design (outside of the DUT in HED)
        instance
verilog_top_block_external_as_text.INST_verilog_block_external_as_text_v use
testLib_external_IP.verilog_block_external_as_text:v;
        instance
verilog_top_block_external_as_text.INST_verilog_block_external_as_text_vams
use testLib_external_IP.verilog_block_external_as_text:vams;
endconfig
```

- For cell binding, that is, when all instantiations of module block_in_virtuoso use the HED configuration:


© 2023 Cadence Design Systems, Inc.

```
config myconfig_verilog_top;
        // design top level (DoT)
        design testLib.verilog_top_block_external_as_text;
        // use of the HED configuration for all instantiations of module
"block_in_virtuoso"
        cell block_in_virtuoso use
cds_amsconfiglib.block_in_virtuoso:amsconfig;
       // instances of module "verilog_block_external_as_text" which are part
of the DoT design (outside of the DUT in HED)
        instance
verilog_top_block_external_as_text.INST_verilog_block_external_as_text_v use
testLib_external_IP.verilog_block_external_as_text:v;
        instance
verilog_top_block_external_as_text.INST_verilog_block_external_as_text_vams
use testLib_external_IP.verilog_block_external_as_text:vams;
endconfig
```

- When a Verilog sub-configuration is used for the BDUs inside an HED configuration, that is, when a block is marked as external text (BDU) in HED, the BDUs block can be configured by using SystemVerilog configurations, an example of which is given below:

```
config myconfig_verilog_top;
        // design top level (DoT)
        design testLib.verilog_top_block_external_as_text;
       // instantiation of the HED configuration for module "block_in_virtuoso"
        instance verilog_top_block_external_as_text.INST_TOP_IN_VIRTUOSO use
cds_amsconfiglib.block_in_virtuoso:amsconfig;
       // instances of module "verilog_block_external_as_text" which are part
of the DoT design (outside of the DUT in HED)
        instance
verilog_top_block_external_as_text.INST_verilog_block_external_as_text_v use
testLib_external_IP.verilog_block_external_as_text:v;
        instance
verilog_top_block_external_as_text.INST_verilog_block_external_as_text_vams
use testLib_external_IP.verilog_block_external_as_text:vams;
       // using verilog sub config for a BDU (verilog_bdu_top)
       instance
verilog_top_block_external_as_text.INST_TOP_IN_VIRTUOSO.INST_verilog_bdu_top_
block_external_as_text use testLib_external_IP_BDU.verilog_bdu_top:config;
endconfig
```

### Working with DSPF files

The Detailed Standard Parasitic Format files (DSPF files) contain the post-layout data, which includes the designed and parasitic devices in a design. These files are created using the parasitic extraction tools. Therefore, the content and format of each DSPF file is dependent on the extraction tool. To run simulations for designs containing devices represented by DSPF


© 2023 Cadence Design Systems, Inc.

files, ADE Assembler needs to consider these components in a way different than the devices with schematic.

For more details, refer to the section Including DSPF files in the Setup in the Virtuoso ADE Assembler User guide.

## Limitations of AMS UNL

The following are the limitations of AMS UNL:

- The HED Reference Verilog Property (`VerilogFile`) must be leaf-level with no hierarchy. The simulator will generate an error if the Verilog files contain hierarchy. `VerilogFile` will eventually be deprecated.
- Verilog-A is considered a true DFII object only and still relies upon shadow and DFII operations.
- It is recommended to update all legacy `Verilog In`-generated standard cell digital libraries, which have old `pc.db` file formats with UDP instance "I" syntax and explicit library bindings. These libraries need `pc.db` files to be recreated, or created if missing, using xmvlog `-use5x` compilations to given proper `pc.db` file format.
- To support a global set of intended parameters for exclusion, you must use the following variable:

  ```
  envSetVal("ams.netlisterOpts"  "amsExcludeParams" 'string  "<param_list>")
  ```

- HED with `pc.db` no longer supports built-in analog primitives (from `analogLib`) structurally instantiated within VerilogAMS text views. Binding cannot be altered on these primitives so they are removed from the `pc.db` file and rely on default Spectre bindings.
- Specification of `modelpath`, `ncuse5x`, and `cdslib` options is not supported in the `hdl.var` file. The following error messages are generated if these options are specified in the `hdl.var` file:

  ```
  ERROR (AMS-2520): Cannot generate the netlist because the 'modelpath' option
  has been specified in the hdl.var file. The modelpath specification in the
  hdl.var file has been deprecated and is not supported in the AMS Unified
  Netlister flow.

  Remove the modelpath option from the hdl.var file and specify it using Setup -
  Model Libraries option and regenerate the netlist.

  ERROR (AMS-2521): Cannot generate the netlist because the 'ncuse5x' option has
  been specified in the hdl.var file. Remove the option from the hdl.var file and
  use xrun to recompile the cell. Next, reference its library for design
  elaboration using the -reflib option in the xrun option form.

  ERROR (AMS-2522): Cannot generate the netlist because the 'cdslib' option has
  been specified in the hdl.var file. The cdslib specification in the hdl.var
  file has been deprecated and is not supported in the AMS Unified Netlister flow.
  Remove the cdslib option from the hdl.var file and regenerate the netlist.
  ```


© 2023 Cadence Design Systems, Inc.

- When importing or editing SystemVerilog, Verilog, or VHDL text source using the DFII text editor, the usual OA shadow creation errors may arise with the content of the text source code and not allow the shadow to be properly created. This is fine with UNL usage since the shadow can be fully ignored now. As long as the `pc.db` is properly created during this step with xmvlog `-use5x`, the text view is ready for UNL netlisting.
- When using inherited connections with HED occurrence and/or inherited *Library List*/*View List* bindings, `cds_variants` will be created internally and used. A problem arises where the parent `inhconn netSet` value does not propagate down the child hierarchy to override child net expressions; therefore, the default property values will be used and will influence simulation results adversely.
- AMS UNL uses the core OSS Netlister and allows a legacy `OSSHNL-907` (warning) to slip through unnoticed by potentially removing cells from the netlist and affecting the resolved bindings at elaboration. This is commonly seen when you do not resolve *red* bindings in the HED and intend those bindings to resolve to external HDL (BDU).

  For example, you may see the following error message in the CIW during netlisting:

  ```
  WARNING (OSSHNL-907): Instance 'I0' of cellview 'invLib/INVCHAIN/
  schematic' in config 'invLib/INVCHAIN_TB/config' has an invalid
  binding. It will be ignored during netlisting.
  ```

  The default value of `hnlInvalidBindingAction` is `error`. Specifying this variable with it's default value in the local `.simrc` file causes AMS UNL to report these errors on such invalid bindings and stop netlisting for you to resolve.
- A change in the text-based 'generate' block construct behavior has taken place with UNL. Partial WDU generate access was visible in HED but changing cellviews below the 'generate' block through HED did not work properly. Generate blocks are not evaluated fully until elaboration, therefore, compiling them and having a populated `pc.db` gives a false static initial state (seen in HED 5x). As a result, the `EXNEUS` errors generated when `pc.db` files are populated with generate instance hierarchy. The bind file is populated using `pc.db` files, therefore, the config card for the generate hierarchical cells gives a false impression that it will be used, however, xmelab determines the final outcome. For UNL usage, `pc.db` files will have generate hierarchical instances removed so `EXNEUS` errors are resolved and proper BDU bindings take place below the parent cellview.

  Cells with generate blocks must have their `pc.db` files updated to reflect this change. To do this, either recompile those cells into their master libraries with xmvlog/xmvhdl `-use5x`, or open the cell in the *Library Manager* and type `:wq!` in the editor window. If the libraries are not writable, check out the cell using DM, make the required changes, and check the cell back into DM.

  If a text cellview contains text hierarchy which has both a generate block with hierarchy and some other non-generate block text hierarchy structure, the parent-level cellview


© 2023 Cadence Design Systems, Inc.

containing both can be compiled into 5x and configured as WDU to allow further WDU hierarchical configuration of the non-generate block text hierarchy. In this case, the generate block and it's hierarchy will be missing from HED visibility anyway since the `pc.db` was recreated to remove the generate block from possible misguided settings in HED. This way, you can still bind to text hierarchy which is not of generate block construct type.

**Note:** With this `pc.db` change, HED configuration bindings will have to be altered for the parent cells that contain generate blocks with hierarchy. These cells must be bound as BDU digitalText with the HED *External HDL* binding mechanism. (This will allow them to auto-bind to digital versions of the cells using the xrun default binding engine). With this required change to BDU bindings, you must make available all generate block parent cell and generate block hierarchical cell source code available using xrun `-y`, `-v`,`-f` options.

- Using the `CDS_ALT_NMP` environment variable for VHDL case-sensitivity handling could change the NMP map algorithm. Cadence recommends that you do not use the following to preserve the case of entity and port names:

  ```
  CDS_ALT_NMP=match
  ```

- Using the xmvhdl 5x option `-keepcase4use5x` in legacy DFII VHDL flows will cause issues with the following scenario of VHDL case-sensitivity and mixed-language (verilog/ vhdl) cellviews:

  ```
  inv.vhd:
  entity \InV\     -- escaped entity name
  end entity
  ```

  The 5.x LCV directory structure looks like the following:

  ```
  library/InV/view
  ```

The netlister generates text inputs like the following:

```
-makelib -specificunit "\InV\" inv.vhd -endlib
```

This works correctly in UNL.

If the entity name is not escaped (\\) while using `-keepcase4use5x`, it results in the following 5.x directory structure: `library/InV/view`

The netlister generates the textInputs file and looks like the following:

```
-makelib -specificunit "\InV\" inv.vhd -endlib
```

This does not work in UNL and results in following error:

```
xmvhdl_p: *W,SPUNFW: specific library unit was never processed 'InV'
```

It is recommended that you escape ( \\ ) the entity name if you need to keep the case sensitivity intact for DFII usage with a mixed language (both Verilog and VHDL) use of


© 2023 Cadence Design Systems, Inc.

cellviews. The xmvhdl compiler always converts the cell name to lower case while xmvlog is case sensitive so it may store the cell in a different 5x directory. This causes a problem since this can result in the VHDL information being stored in a different directory than the Verilog module. HED (and other DFII tools) will not be able to find the VHDL information in the same location as the Verilog information (it will be treated as a different cell).

- If Spectre user-defined functions (UDFs) are used only in the context of Spectre, then it works fine in the AMS flow. However, if the Spectre UDFs are used in the context of Verilog-AMS namespace, then you need to manually convert the Spectre UDFs into a Verilog-compatible macro and include those within the `netlist.vams` file.

  If required, macro conversion should be done manually just like any other cross-language function/macro conversion step. That is, a VHDL macro requires conversion to a Verilog macro to be used in that namespace. The case is the same for a Spectre function being used in a Verilog-AMS namespace. This flow requires you to perform the UDF to macro conversion manually, and then include the `macros.vams` header file in the *Include files* field under the *Netlister* tab of the *AMS Options* form.

**Should an instance that is bound to a symbol view be netlisted in Spectre syntax (using spectre siminfo) or in Verilog syntax?**

By default, if an instance in your design configuration is bound to a symbol view, the instance is considered digital, and hence netlisted in Verilog syntax.

If you want a specific instance to be considered analog, and netlisted in Spectre syntax using Spectre siminfo, do the following:

1. Open your configuration in Hierarchy Editor.
2. Choose *View – Tree*.
3. Right-click the instance and choose *Set Instance View – Spectre*.

If you want all instances that are bound to a symbol view to be considered analog, do the following:

1. Choose *Simulation – Options – AMS Simulator.*

   Th AMS Options form appears. for more informati, see Setting up AMS Options on page 1012.
2. Click the *Netlister* tab.
3. Add `symbol` in the *Netlist using spectre CDF simInfo* field.


© 2023 Cadence Design Systems, Inc.

### Netlisting of inherited connections

Inherited connections allow you to selectively override the global signals in your design. This allows you to use multiple power supplies in a single design. If you want to implement separate power supplies (analog and digital, for example, or `+3 V` and `+5 V`) in a hierarchical design, you can assign net expressions to those global signals whose defaults you might want to override. Then you can use `netSet` properties to specify the new values of the signals. For more information about inherited connections, see the *Inherited Connections Flow Guide* and the *Virtuoso Inherited Connections Tutorial*.

The `netSet` properties are netlisted as `cds_net_set` attributes and evaluated by the elaborator. For example, the following `netSet` properties

| Property | Value |
|---|---|
| `vdd` | `3.3v!` |
| `xground` | `[@new_ground:%:gnd5!]` |

will be netlisted as:

```
(*
integer cds_net_set
[0:1] = {"xground", "vdd"};
integer xground[0:1] = {"new_ground", "cds_globals.\\gnd5!" } ;
integer vdd = "cds_globals.\\3.3v!" ;
*)
```

#### *Important points to note:*

- Use environment variable `UNL_KEEP_ACCURACY` to convert a double type value to a string type, then print it in `dna_assembler` process in netlisting. This conversion makes the date more accurate because a design variable value of type double is not truncated. For example, in `dna_assembler` flow, if a design variable of type double has value `32.0195312500`, it is printed in `%g` format. It prints up to 6 bits of floating point data, therefore, losing accuracy in the value.

  To set this variable in the CIW, use the call:

  ```
  setenv UNL_KEEP_ACCURACY YES
  ```

- To use inherited connections with Verilog A modules, you must add an additional port and specify the inherited connection attributes for each inherited connection.

  For example, in the Verilog A definition shown below:

  ```
  module inhconn_spectre(vdd_out, gnd_out, global_vdd, global_gnd);
     inout vdd_out, gnd_out;
  ```


© 2023 Cadence Design Systems, Inc.

```
   input (* integer inh_conn_prop_name="global_vdd_prop";
            integer inh_conn_def_value="cds_globals.\\vdd! "; *) global_vdd;
   input (* integer inh_conn_prop_name="global_gnd_prop";
            integer inh_conn_def_value="cds_globals.\\gnd! "; *) global_gnd;
   electrical vdd_out, gnd_out, global_vdd, global_gnd;

   ...
endmodule
```

- The ports `global_vdd` and `global_gnd` are extra ports added to use inherited connections.
- The inherited connection attributes for the `global_vdd` port are specified as

  ```
  (* integer inh_conn_prop_name="global_vdd_prop";
  integer inh_conn_def_value="cds_globals.\\vdd! "; *)
  ```

  and the inherited connection attributes for the `global_gnd` port is specified as

  ```
  (* integer inh_conn_prop_name="global_gnd_prop";
  integer inh_conn_def_value="cds_globals.\\gnd! "; *)
  ```

  Where the `inh_conn_prop_name` attribute specifies the name of the `netSet` property, and the `inh_conn_def_value` attribute specifies the value of the `netSet` property.

When the design is netlisted, the inherited connection attributes are printed in the netlist as:

```
wire (*
integer inh_conn_prop_name = "global_gnd_prop";
integer inh_conn_def_value = "cds_globals.\\gnd! ";
*)
\global_gnd_prop_gnd! ;
wire (*
integer inh_conn_prop_name = "global_vdd_prop";
integer inh_conn_def_value = "cds_globals.\\vdd! ";
*)
\global_vdd_prop_vdd! ;
```

and the I7 instance of the `inhconn_spectre` cell is netlisted as:

```
inhconn_spectre  I7 ( vdd_va_spectre , gnd_va_spectre , \global_vdd_prop_vdd!
, \global_gnd_prop_gnd! );
```

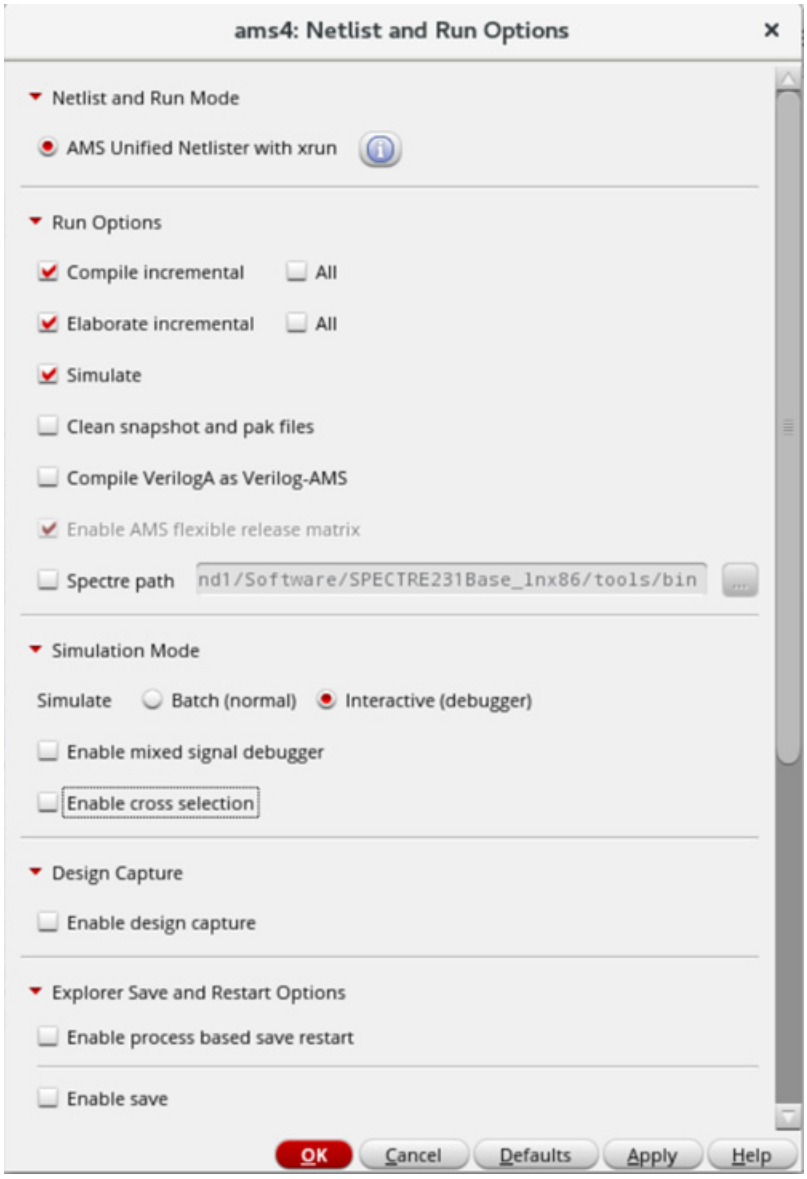

© 2023 Cadence Design Systems, Inc.

# Setting Up the AMS Netlister Options

Use the AMS template in Hierarchy Editor to create the configuration for your top-level design. To set the options required for using AMS UNL:

1. Choose *Simulation – Netlist and Run Options*.

   The Netlist and Run Options form is displayed.




   The *AMS Unified Netlisting with xrun* option is selected by default when a new session is opened.


© 2023 Cadence Design Systems, Inc.

2. In the *Run Options* section, specify the run options for the netlister.
   - *Compile Incremental*

     Select *All* to compile all modules.
   - *Elaborate Incremental*

     Select *All* to elaborate all modules.
   - *Simulate*
   - *Clean snapshot and pak files*
   - *Compile VerilogA as Verilog-AMS*
   - *Enable AMS flexible release matrix*
   - *Spectre path*
3. In the *Simulation Mode* section:
   a. Select the simulation mode as *Batch (normal)* or *Interactive (debugger)*
   b. Select the *Enable mixed signal debugger* check box to view the SimVision debug environment for AMS simulator

      **Note:** This option is available only if the *Enable AMS flexible release matrix* check box is selected.
   c. Select *Enable cross selection* to select an object in Schematic Editor and have the corresponding simulation object selected in SimVision, and vice versa.
4. In the *Design Capture* section, select *Enable design capture* to trace the files that are accessed by a simulation and package them so that the same setup can be run in a different session.
5. In the Explorer *Save and Restart Options* section, select *Enable process based save and restart* to enable the process-based save and restart flow.
6. Select *Enable save* to enable the *Save Options* group box.
7. In the *Save Options* section, specify the options for saving simulation snapshots.

   For more information, see Save Options.
8. Select *Enable restart* to enable the *Restart Options* group box.
9. In the *Restart Options* section, specify the options for restarting a simulation from the saved snapshots.


© 2023 Cadence Design Systems, Inc.

**10.** Click *OK*.

***Related Topics***

Process-Based Save and Restart (PBSR) flow

Saving Simulation Snapshots at Different Time Points

Saving Simulation Snapshots Periodically

Restarting a Simulation from a Saved Snapshot

Netlist and Run Options Form

# Process-Based Save and Restart (PBSR) flow

You can select *Enable process based save restart* in the Netlist and Run Options form to enable the Process- Based Save and Restart (PBSR) flow for Spectre AMS Designer simulation runs. This lets you save simulation snapshots during a transient analysis simulation run and later restart the simulation from a specific snapshot.

Saving simulation snapshots is especially useful for large simulations where you might want to save the simulation state at regular intervals.

This features lets you:

- Save time by restarting simulation from a specific snapshot instead of restarting simulation from time `0`.
- Quickly debug simulation results by verifying only the snapshots in which you are interested.
- Restart from a known good point if simulation fails at any time due to a system failure, power outage, or other issues.

It is recommended to use the *Batch* simulation mode while using the PBSR flow. When you use this flow with the *Interactive(debugger)* simulation mode, AMS prints a statement in the `probe.tcl` file, as shown below, which pauses the interactive simulation at the SimVision MS console.

```
stop -create -time 40ns -absolute -execute "save snapshot_40n -
overwrite" -continue
```


© 2023 Cadence Design Systems, Inc.

If you specify any additional save commands or restart point other than the ones specified in the Netlist and Run Options form, it leads to an unsupported state of the PBSR database.

## Saving Simulation Snapshots at Different Time Points

To save simulation snapshots at different time points:

1. In the *Explorer Save and Restart Options* section of the Netlist and Run Options form, select *Time points*.
2. Specify a prefix for snapshot names in the *Snapshot prefix* field.
3. In the *Save time points* field, specify the time points at which the snapshots must be saved.

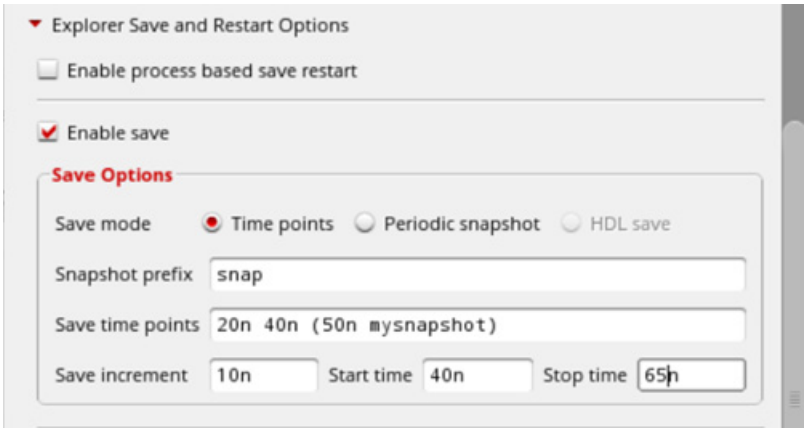
   Note the following:

   - Do not use spaces between the time and the unit of time. The default unit of time is seconds.
   - Multiple values must be separated by blank spaces.
   - To save a snapshot for a time point with a unique name, specify the time point in the following format:

     `(timePoint snapShotName)`

     For example, if you specify the time point as `(50n my50nSnapShot)`, a snapshot named `my50nSnapShot` is saved at `50n`.
4. In the *Save increment* field, specify the time interval at which the snapshots must be saved. In *Start time* and *Stop time*, specify the respective time points at which the snapshots must start and stop.

   For example, let us consider the following setup.





© 2023 Cadence Design Systems, Inc.

In this case, the following four snapshots are saved at `20n`, `40n`, `50n` and `60n`.

- `snap_20n`
- `snap_40n`
- `my50nSnapShot`
- `snap_60n`

No snapshot is saved at `65n` because after `40n`, snapshots are saved only at intervals of `10n`

**Note:** Do not use spaces between the specified value of time and the unit. The default unit is seconds.

5. Click *OK*.
6. Run a simulation.

***Related Topics***

Process-Based Save and Restart (PBSR) flow

Saving Simulation Snapshots Periodically

Pre-requisites for Restarting a simulation from a Saved Snapshot

Netlist and Run Options Form

## Saving Simulation Snapshots Periodically

To save simulation snapshots periodically:

1. In the *Explorer Save and Restart Options* section of the Netlist and Run Options form, select *Periodic snapshot*.
2. Specify the base name for the snapshot in the *Snapshot base name* field.
3. In the *Strobe time* field, specify the time interval between saving snapshots.
4. Select one of the following options to specify the number of snapshots to be saved for each simulation run.
   - *Max Number*
   - *All*


© 2023 Cadence Design Systems, Inc.

5. Specify the time unit to be used in the *Snapshot name time unit* field.
6. Click OK.
7. Run a simulation.

***Related Topics***

Process-Based Save and Restart (PBSR) flow

Saving Simulation Snapshots at Different Time Points

Pre-requisites for Restarting a simulation from a Saved Snapshot

Netlist and Run Options Form

## Pre-requisites for Restarting a simulation from a Saved Snapshot

You can restart a simulation from an existing snapshot if it meets one of the following criteria:

- The snapshot was created using the same design that is currently being used.
- Only the options that do not affect the topology, such as `reltol`, `abstol`, `errpreset`, `method`, and `stop`, in the Transient Options form were changed since the snapshot was created.
- The model files have not been changed since the snapshot was created.
- The solver used to save the snapshot is the same as that being used for the simulation. For example, if a snapshot was saved when `spectre` was set as the solver, you must restart simulation using the snapshot only with `spectre` is as the solver.

## Restarting a Simulation from a Saved Snapshot

To restart a simulation from a saved snapshot

1. Select the *Enable Restart* check box in the Netlist and Run Options form to enable the *Restart Options* group box.

   The fields *Test name* and *history* display the name of the test and the last history associated with the current maestro session, respectively. These fields are not editable.
2. From the *Snapshot name* drop-down list, select the snapshot from which you want to restart the simulation.


© 2023 Cadence Design Systems, Inc.

3. Click *OK*.
4. Run a simulation.

When the simulation completes, a waveform from time `0` to the specified stop time is displayed. This waveform is plotted even if the simulation was started from a later snapshot, and the snapshots were saved during different simulation runs.

### *Related Topics*

Process-Based Save and Restart (PBSR) flow

Saving Simulation Snapshots at Different Time Points

Saving Simulation Snapshots Periodically

Netlist and Run Options Form


© 2023 Cadence Design Systems, Inc.

## Netlist and Run Options Form

The Netlist and Run Options form lets you specify the options required for using AMS UNL.

| Field | Description |
|---|---|
| ***Netlist and Run Mode*** | Specifies whether you have Xcelium or INCISIVE in your installation path. |
| *AMS Unified Netlister with xrun* | Opens the AMS UNL documentation in the Doc Assistant.<br>The options displayed in the form depend on the installation path. If you have Xcelium in the installation path, Xcelium commands, such as xrun are displayed. If your installation path has INCISIVE, INCISIVE commands, such as irun are displayed.<br>**Note:** In case the `AMSXLMCOMPATIBLE` environment variable is set, INCISIVE commands are displayed in the form even if you have Xcelium in your installation path. |
| ***Run Options*** | Specifies the run options. |
| *Compile Incremental* | Compiles just the modules that have been edited since the last compilation. |
| *All* | Compiles all modules. |
| *Elaborate Incremental* | Elabortaes just the modules that have been edited since the last elaboration. |
| *All* | Elaborates all Models. |
| *Simulate* | Simulates the design.<br>Note the following:<br>■ You can choose to only compile, elaborate or simulate the design, although by default all three are selected.<br>For example, if you select only *Elaborate*, the *Simulation – Netlist* command only elaborates the design. Compilation and elaboration can be incremental or for the whole design together.<br>■ A simulation will fail if you choose both *Compile* and *Simulate*. You would need to either deselect *Simulate* or select *Elaborate* as well. |


© 2023 Cadence Design Systems, Inc.

| Field | Description |
|---|---|
| *Clean snapshot and pak files* | Deletes any existing simulation snapshot and .pak files before running the simulation.<br><br>This option is available for the xrun run mode only. |
| *Compile VerilogA as Verilog-AMS* | Indicates that the AMS simulator should compile VerilogA as Verilog-AMS.<br><br>This option is available for the xrun run mode only.<br><br>**Note:** On selecting the *Netlist using Spectre* check box is selected in the AMS Options form, this option gets disabled. |
| *Enable AMS flexible release matrix* | Enables the AMS flex flow.<br><br>This check box is greyed out and selected automatically with Xcelium 21.02 or a higher release version. |
| *Spectre path* | Specifies the path to the Spectre version to be used with AMS flex flow. This lets you use different versions of Spectre for the Xcelium build you are using.<br><br>**Note:** This option is available only when *Enable AMS flexible release matrix* is selected. |
| ***Simulation Mode*** | Specifies the simulation options. |
| *Simulate* | Specifies the simulation mode.<br><br>■ *Batch (normal)*: Plots signals in Virtuoso Visualization and Analysis XL.<br><br>This is the default mode.<br><br>■ *Interactive (debugger)*: Launches the complete SimVision MS debug environment on top of the current design.<br><br>When this mode is selected, set the DISPLAY shell environment variable in the `hostname:portNumber` format so that the SimVision environment window is displayed. To ensure that the variable has the value in the correct format, set the `addHostNameToDisplayPort` environment variable to `t`.<br><br>**Note:** When the host mode is set to distributed, the radio button for interactive mode is disabled and batch mode is enabled (only if Interactive mode was selected before). |


© 2023 Cadence Design Systems, Inc.

| Field | Description |
|---|---|
| *Enable mixed signal debugger* | Displays the SimVision MS debug environment for AMS simulator.<br>Cadence recommends that you enable the process-based save restart flow through this check box instead of specifying the option `-process_save` as an additional argument.<br>**Note:** This option is available only if the Enable AMS flexible release matrix check box is selected. |
| *Enable cross selection* | Lets you select an object in Schematic Editor and have the corresponding simulation object selected in SimVision, and vice versa. This helps in debugging design errors. |
| ***Design Capture*** | Specifies the options for the Design Capture utility. |
| *Enable design capture* | Traces the files that are accessed by a simulation and packages them so that the same setup can be run in a different session.<br>This feature can be used only when:<br>■ The *Batch (normal)* simulation mode is selected<br>■ The settings in the *Explorer and Restart Options* group box are not being used<br>■ The settings in the *Run Options* group box required to compile, elaborate, and simulate the design are selected |
| ***Explorer Save and Restart Options*** | Specifies the save and restart options for ADE Explorer.<br>**Note:** When you switch to ADE Assembler, the name of this section changes to *Assembler Save and Restart Options*. |
| *Enable process based save and restart* | Enables the process-based save and restart flow that lets you save the entire simulation setup, and facilitates a smoother restart operation, so that the future simulations can continue from that point.<br>While restarting the simulation from a saved setup, the following points must be kept in mind:<br>■ The circuit topology cannot be changed.<br>■ The product versions used while creating the saved setup must not be altered while restarting the simulation. |
| *Enable save* | Enables the *Save Options* group box. |


© 2023 Cadence Design Systems, Inc.

| Field | Description |
|---|---|
| ***Save Options*** | Specifies the options for saving simulation snapshots. |
| *Save mode* | Specifies the mode for saving simulation snapshots.<br>■ *Time points*: Saves the snapshots at specific time points.<br>■ *Periodic snapshot*: Saves the snapshots periodically.<br>■ *HDL save*: Represents the system function `$save` that can be used in the Verilog, Verilog-AMS, and SystemVerilog modules. This enables you to set up the process-based save and restart operation in relation to the design or testbench events specified in the Verilog code.<br>**Note:** Ensure that the *HDL save* check box is selected when saving a snapshot using `$save` in a vams file |
| *Snapshot prefix* | Specifies the prefix for the snapshot name.<br>For example, if you specify the prefix as `mySnap`, snapshots are saved with names such as `mySnap_40n`, where `40n` indicates the timepoint at which the snapshot was saved.<br>The default prefix is `snapshot`. |
| *Save time points* | Specifies the time points at which snapshots need to be saved.<br>Note the following:<br>■ Do not use spaces between the time and the unit of time. The default unit of time is seconds.<br>■ Multiple values must be separated by blank spaces.<br>■ To save a snapshot for a time point with a unique name, specify the time point in the following format:<br>`(timePoint snapShotName)`<br>For example, if you specify the time point as (`50n my50nSnapShot`), a snapshot named `my50nSnapShot` is saved for time point `50n`. |
| *Save increment* | Specifies the time interval at which the snapshots must be saved. |
| *Start Time* | Specifies the starting time point for the snapshots to be taken.<br>**Note:** The default unit of time is seconds. Ensure that you do not use spaces between the time and the unit. |


© 2023 Cadence Design Systems, Inc.

| Field | Description |
|---|---|
| *Stop Time* | Specifies the stopping time point for the snapshots to be taken.<br><br>**Note:** Do not use spaces between the time and the unit. The default unit of time is seconds. |
| *Snapshot base name* | Specifies the base name for the snapshot.<br><br>The snapshot base name must be alphanumeric with no spaces, underscores, or special characters. |
| *Strobe time* | Specifies the time interval between saving snapshots. It can be a numerical value, followed by the unit or a percentage value.<br><br>On adding a percentage value, the time interval between saving snapshots is then calculated as the specified percentage of the value of time given in the *Stop Time* field. The value for this field can be specified in the Choosing Analyses form for transient analysis.<br><br>For example, on specifying the stop time as `1u` and strobe time as `15%`, the equivalent strobe time would be `150n`.<br><br>**Note:** Do not use spaces between the time and the unit of time. The default unit is seconds. |


© 2023 Cadence Design Systems, Inc.

| Field | Description |
|---|---|
| *Snapshots per run* | Specifies the number of snapshots to be saved in one simulation run.<br>■ *Max Number*: Lets you specify the maximum number of snapshots to be saved in one simulation run.<br>For example, if the value for the *Max Number* field is set to `4`, *Strobe time* is `20n`, and the *Snapshot name time unit* is `us`, the screenshots are saved as follows:<br>`mySnapShot1run1`<br>`mySnapShot1run2`<br>`mySnapShot1run3`<br>`mySnapShot1run4`<br>The specified number is applicable only if the *All* option is deselected.<br>■ *All*: Specifies that the all the snapshots taken during the time interval specified in the *Strobe time* field must be saved. |
| *Snapshot name time unit* | Specifies the time unit to be used in the snapshot name.<br>For example, if you specify the strobe time as `0.25u`, and the stop time as `1u`, a screenshot named `mySnapShot_250_NS_run1` is saved.<br>The changes made to this field are applicable only if the *Save all snapshots* check box is selected. |
| *Enable restart* | Enables the *Restart Options* group box.<br>When this check box is selected, the *Compile incremental* and *Elaborate incremental* fields are disabled because netlisting, compilation and elaboration are disabled when you restart simulation from a saved snapshot. This ensures that existing snapshots are not overwritten. |
| ***Restart Options*** | Specifies the restart options. |
| *Test name* | Displays the name of the test for which the Netlist and Run Options form is opened.<br>This is not an editable field. |


© 2023 Cadence Design Systems, Inc.

| Field | Description |
|---|---|
| *History name* | Displays the name of the last history associated with the current maestro session.<br><br>**Note:** In ADE Assembler, this field displays the history select through the Reference Histories form.<br><br>This is not an editable field. |
| *Snapshot name* | Lets you select the snapshot from which the simulation is to be restarted. |
| *Add* | Lets you add the select snapshot to the *Snapshot Sweep Information* table.<br><br>**Note:** This button is available only when ADE Assembler is used. |
| *Snapshot Sweep Information* | Displays the information for the selected snapshots in tabular format.<br><br>**Note:** This table is available only when ADE Assembler is used.<br><br>It has the following columns:<br><br>■ *Enable*: Enables or disables the entry for a snapshot.<br><br>■ *Test*: Specifies the name of the test associated with the given snapshot.<br><br>■ *History*: Displays the name of the history associated with the given snapshot.<br><br>■ *Snapshot*: Displays the name of the selected snapshot.<br><br>■ *tclFile*: Lets you specify a Tcl file for the given snapshot. |

***Related Topics***

Setting Up the AMS Netlister Options

Process-Based Save and Restart (PBSR) flow

Saving Simulation Snapshots at Different Time Points

Saving Simulation Snapshots Periodically

Restarting a Simulation from a Saved Snapshot


© 2023 Cadence Design Systems, Inc.

## Simulating the Design Using the xrun Command

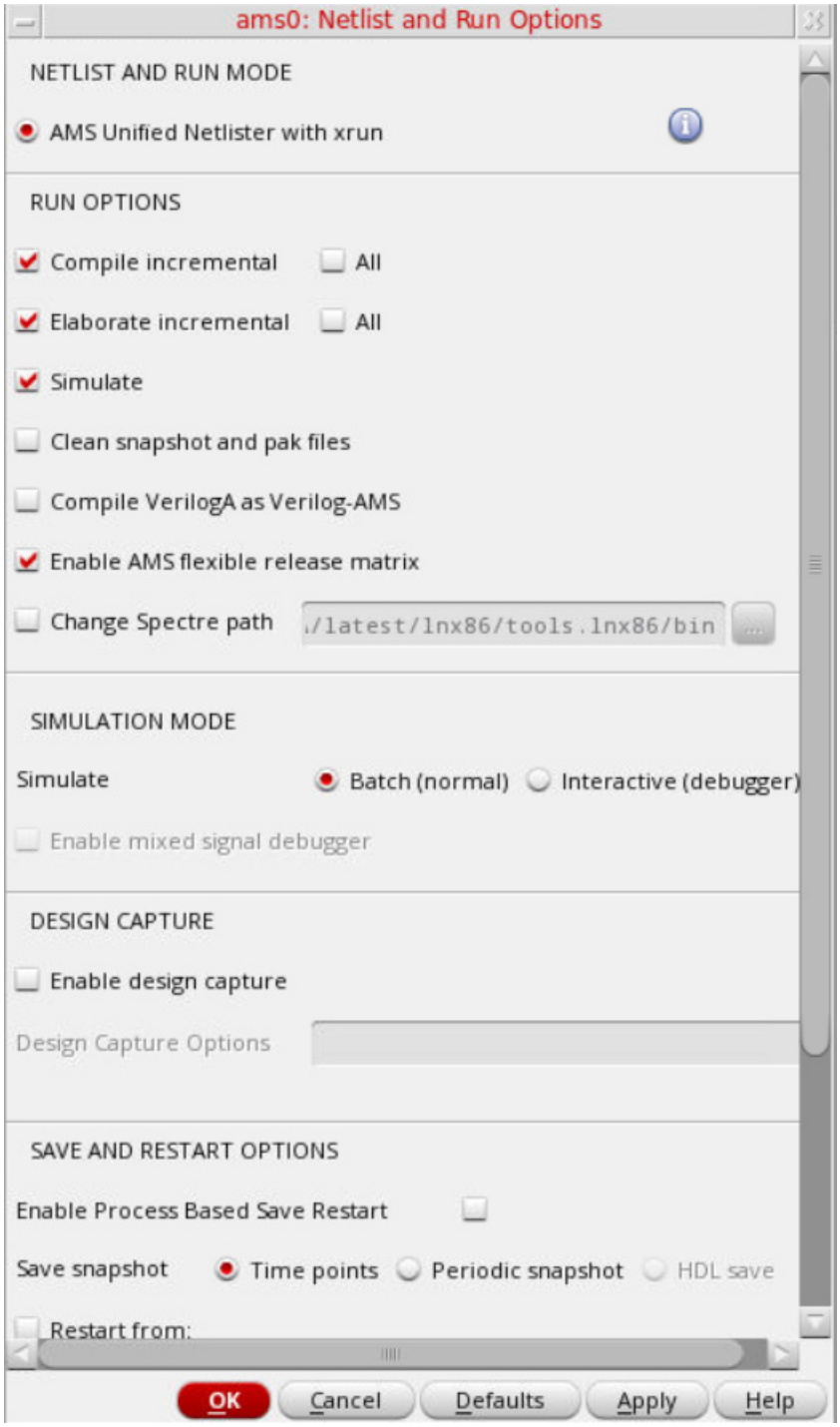
AMS UNL supports simulation using the `xrun` command of the Spectre AMS Designer and Xcelium Mixed-Signal simulator.

The xrun command compiles text using a file extension parser. For example, files with the `.v` extension are compiled using the `xmvlog` Verilog compiler, files with the `.vhd` extension are compiled using the `xmvhdl` VHDL compiler, and files with the `.vams` extension are compiled using the `xmvlog` Verilog compiler with the `-ams` argument. After the input files are compiled, xrun automatically starts `xmelab` to elaborate the design and then uses `xmsim` to simulate the design. For more information about the xrun command, see the *Spectre AMS Designer and Xcelium Simulator Mixed-Signal User Guide*.

**Note:** You need to install XCELIUM 17.10 s007 or a later release to use the xrun command with the AMS UNL flow.


© 2023 Cadence Design Systems, Inc.

You can specify the options for running `xrun` using the AMS Options form.





© 2023 Cadence Design Systems, Inc.

### Netlist and Run Command

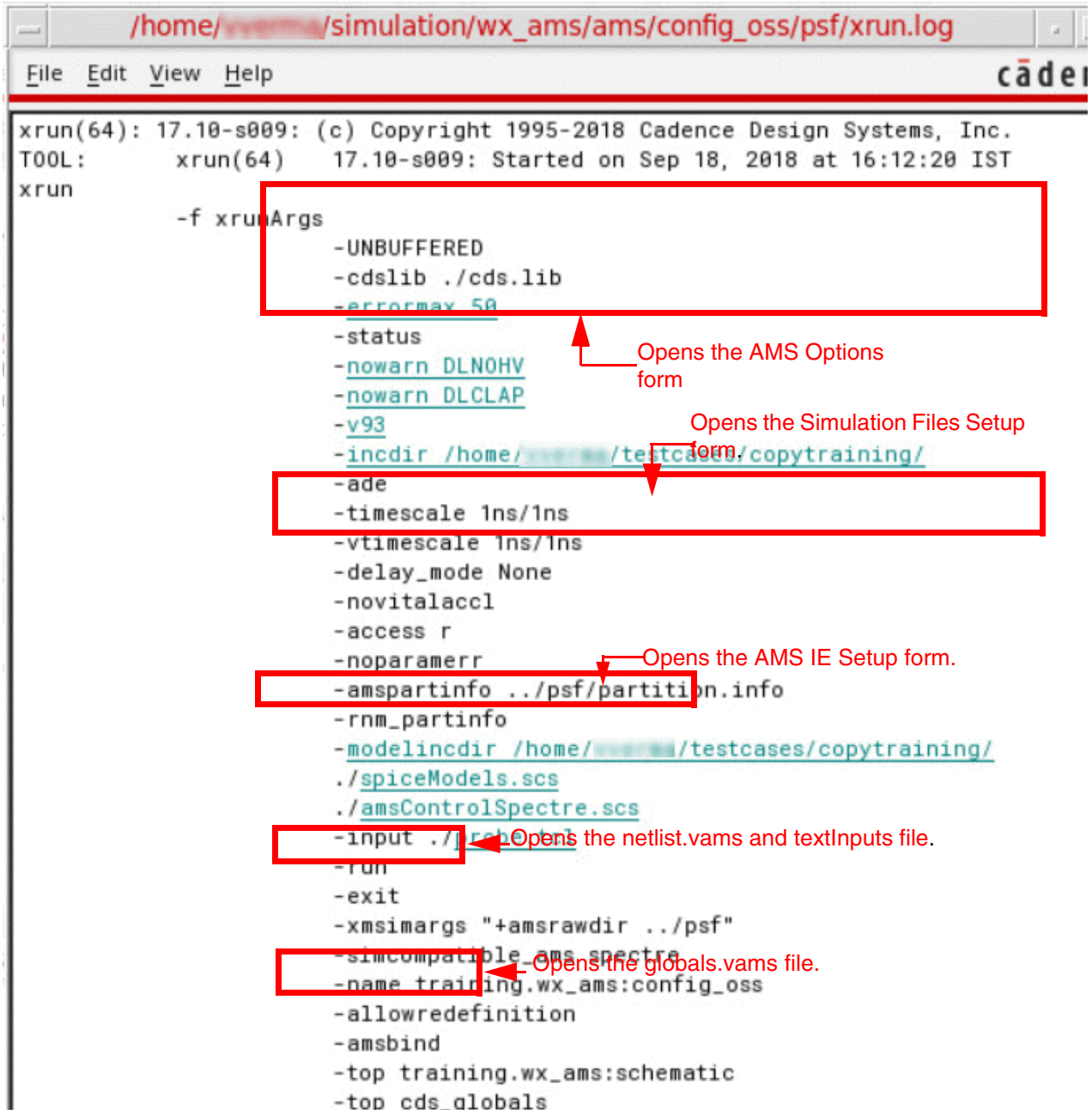
The *Netlist and Run* command generates the xrun command using the options specified in the AMS Options form.

After running the *Netlist and Run* command, you can view the log file named `xrun.log` using, *Simulation – Output Log – xrun Log*. All the errors generated for the compiler, elaborator and AMS Designer simulator can be viewed in this log file.

When you run simulation using AMS UNL, the `xrun.log` file contains hyperlinks to various components for better interaction with the design. You can use these hyperlinks to open the AMS Options form, Simulation Files Setup form, IE Setup form, Hierarchy Editor, and various .vams output files. To disable these hyperlinks from appearing in the `xrun.log` file, set the `UNL_HYPERLINK_LOG` environment variable to `NO` in CIW.

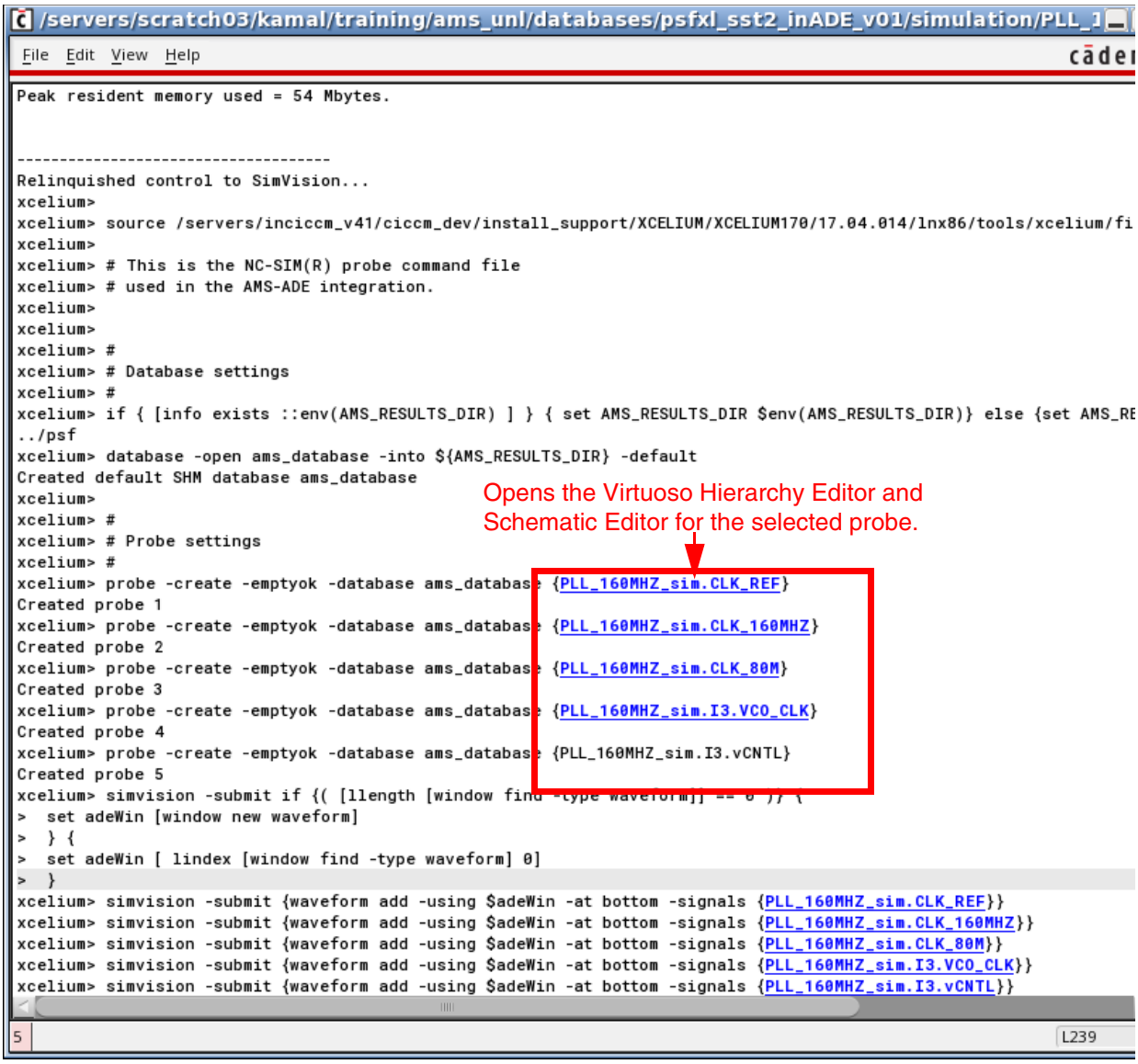

© 2023 Cadence Design Systems, Inc.

Refer to the following figures to know the available hyperlinks and their functions:





© 2023 Cadence Design Systems, Inc.




## Debugging Verilog-A Instances in AVUM flow

To debug verilog instances in AVUM flow, perform the following steps.

1. In the Netlist and Run Options form:
   a. Select *Compile Verilog-A as Verilog-AMS* in the *RUN OPTIONS* group box
   b. Set the simulation mode to Interactive(debugger)


© 2023 Cadence Design Systems, Inc.

**c.** Select *Enable mixed signal debugger*

2. In the AMS Options form, select *Read* from the *Access visibility* drop-down list.
3. Run a simulation.
4. In the SimVision Design Browser, select the Verilog-A instance you want to plot.

You can then use the SimVision MS to create a TCL probe and plot the instances.

***Related Topics***

How to probe and debug Verilog-A variables, nets, ports using Tcl in AVUM flow (AMS Designer in ADE flow)

## Saving or Plotting Selected Voltages or Currents for AMS Simulation

For AMS simulation, you can use the schematic window to select the nodes or terminals for instances that have a schematic view.

You can use the Hierarchy Editor to select the nodes or terminals for instances:

- That have a Verilog, VerilogA, VerilogAMS, VHDL, VHDLAMS or SPICE-based text view
- That are bound to a Verilog, VerilogA, VerilogAMS, VHDL, VHDLAMS or SPICE-based text file using the `sourcefile` or `verilogfile` property in Hierarchy Editor.

To specify the nodes and terminals for an instance, which has a schematic view, to be saved or plotted, do the folowing:

1. Choose *Outputs – To Be Saved – Select On Design*.
2. Choose *Outputs – To Be Plotted – Select On Design*.

   The Schematic window appears.

   **Note:** If the top-level design is a text design, the *Choose an instance to descend* form appears instead of the Schematic window. Select an instance in the list and click *OK* to open the schematic for the instance. You can also double-click an instance to open the schematic.

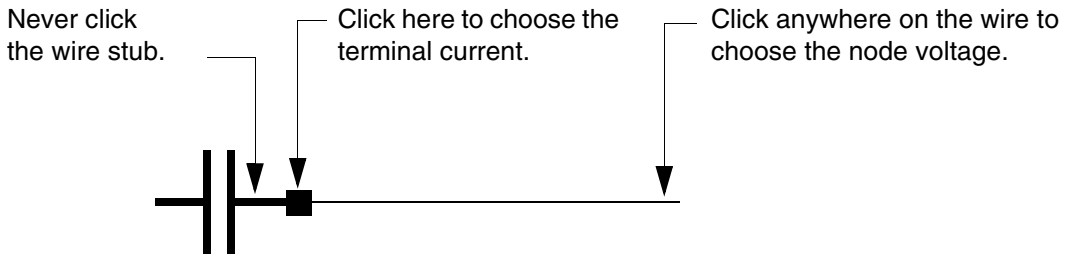
3. In the Schematic window, choose one or more nodes or terminals.

   The system circles pins when you choose a current and highlights wires when you choose a net.

   - ❑ Click an instance to choose all instance terminals.


© 2023 Cadence Design Systems, Inc.

- Click the square pin symbols to choose currents.
- Click wires to choose voltages.
- Click and drag to choose voltages by area.




4. Press the *Esc* key when you are done.

   Your selections are added in the *Outputs* list in the simulation window.

To specify the nodes and terminals for an instance, which has a text view, to be saved or plotted, do the folowing:

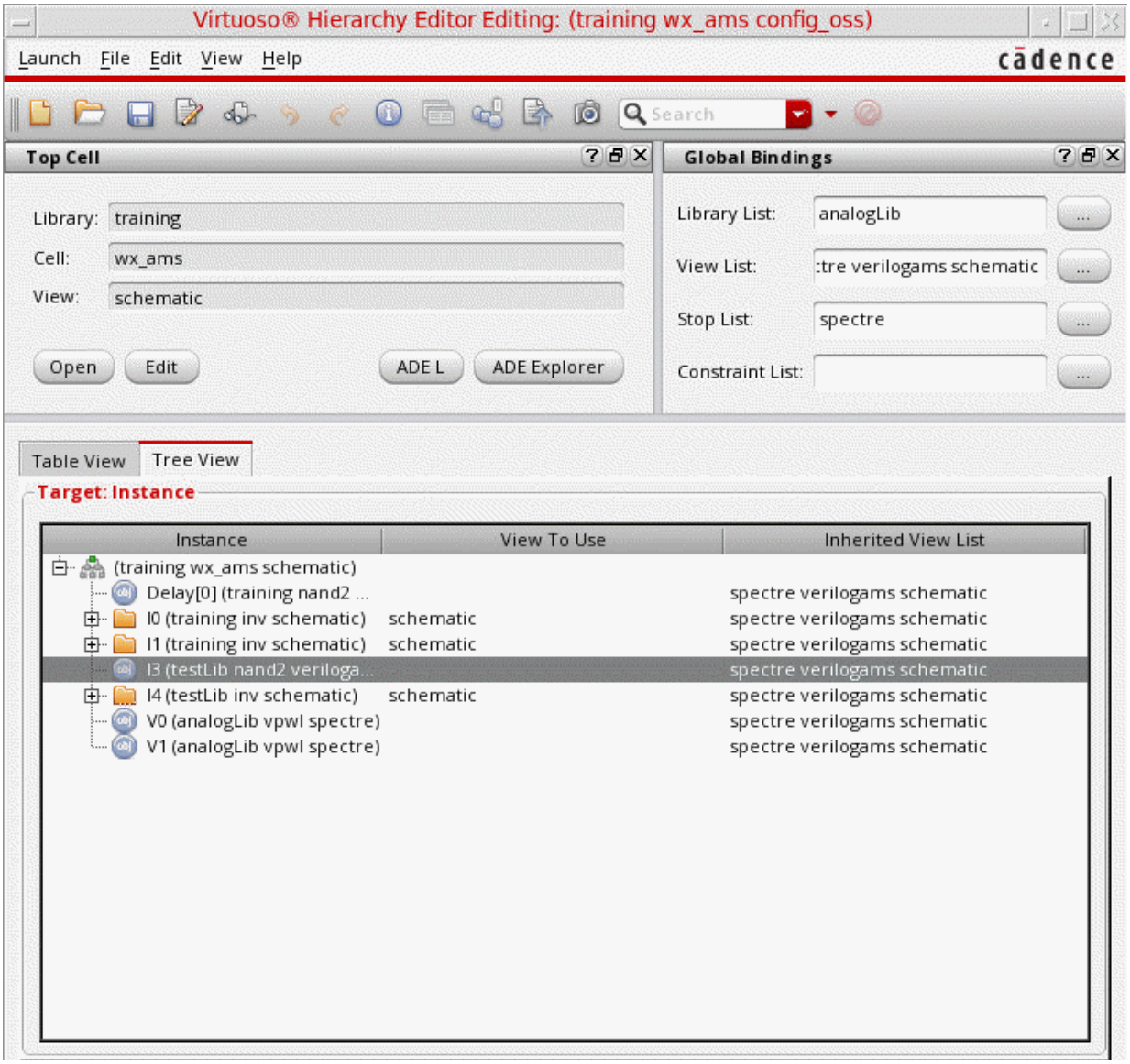
1. Choose *Outputs – To Be Saved – Select From HED*
2. Choose *Outputs – To Be Plotted – Select From HED*.

   The Hierarchy Editor appears displaying the instances in the tree view.

   **Note:** You can select nodes and terminals for instances only in the tree view in Hierarchy Editor. If you are in the table view in Hierarchy Editor, you cannot select nodes and terminals for instances.


© 2023 Cadence Design Systems, Inc.

**3.** Click the instance with the text view.




In the above figure, instance `I3` is selected in Hierarchy Editor.


© 2023 Cadence Design Systems, Inc.

The Select Net/Term to Save/Plot form appears.

4. Do one of the following:

   - Select *net* to save or plot voltage.
   - Select *term* to save or plot current.

5. Select the node or terminal for the instance from the *Choose net/term* cyclic field.

   You can click the *View* button to view the module for the instance in a text editor.

   **Note:** The terminals of a primitive instance will not be displayed in the *Choose net/term* cyclic field because primitive instances do not have a text view. You can enter the terminal name in the *Enter net/term* field using the format:

   ```
   I3/Y
   ```

   Where `/I3` is the full instance name and `Y` is the terminal name.

6. Do one of the following:

   - Click *Apply* to add the node or terminal in the *Outputs* list in the simulation window. Select another net or term for the current instance, or perform steps 3 to 5 to select a net or term for another instance.
   - Click *OK* to close the Select Net/Term to Save/Plot form and add the node or terminal in the *Outputs* list in the simulation window.

   If you click *OK* or *Cancel* and then want to select nodes or terminals from Hierarchy Editor, you must again choose *Outputs – To Be Saved – Select From HED* or *Outputs – To Be Plotted – Select From HED* in ADE Explorer.

*Important*

In flex mode, -`amsformat` is ignored. The default waveform format is a unified database with analog data in `PSFXL 1.1` format and digital data in `SST2` format. To change the waveform format for analog users, use the `rawfmt` Spectre option or the


© 2023 Cadence Design Systems, Inc.

`-format` Spectre command-line argument. In such cases, unified database is disabled and the results will be stored in two different files, one for analog results using the specified Spectre format and one for digital data in `SST2` format.

When you enable the flex mode without PSF XL as the output format, you see a message as follows:

AMSD flex flow is enabled but the recommended PSF XL unified database format is not enabled.

The message box offers the following choices:

- Change the output format to PSF XL. In this case, click *Yes*.
- Run the AMSD flex flow with the current output format. In this case, click *No*.

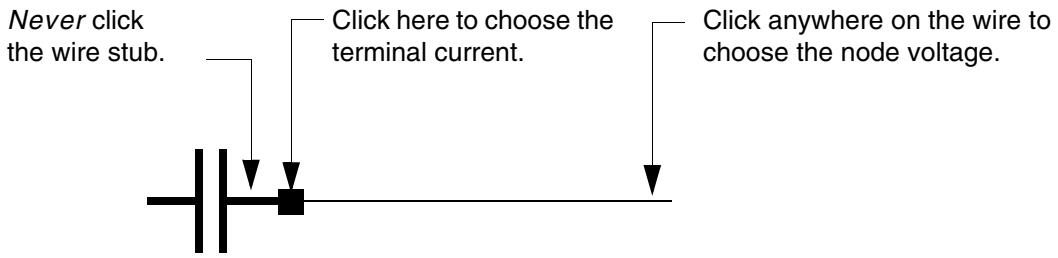
## Adding a Node or Terminal to a Set

To add a node or terminal to the saved or plotted sets:

1. Choose *Outputs – To Be – Select On Design* commands in the Simulation window.
2. In the Schematic window, choose one or more nodes or terminals.

   The system circles pins when you choose a current and highlights wires when you choose a net.

   - Click the square pin symbols to choose currents.
   - Click wires to choose voltages.
   - Click and drag to choose voltages by area.




3. Press the `Esc` key when you finish.

To select nodes and terminals in lower-level schematics to be plotted or saved,

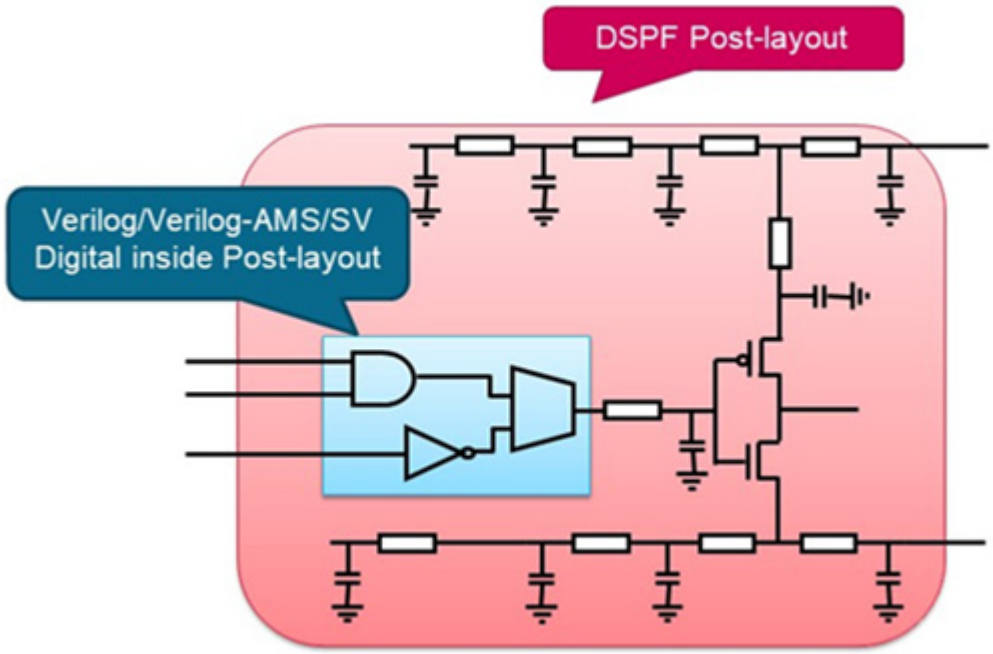
1. In the Simulation window, choose *Outputs – To Be – Select On Design*.


© 2023 Cadence Design Systems, Inc.

2. In the Schematic window, choose *Design – Hierarchy – Descend Edit* and click an instance.
3. Click *OK* in the form that appears.
4. In the Schematic window, choose one or more nodes or terminals.
5. Press the `Esc` key when you finish.

# DSPF-in-the-Middle Flow for AMS Simulations

The DSPF-in-the-middle flow allows the use of a digital module inside DSPF and speeds up mixed-signal post-layout simulation. This enables running verification in the early stages.




To enable the DSPF-in-the-middle flow for ADE Explorer or ADE Assembler, select *Using HED Config for DSPF Blackbox* in the Simulation Files Setup form for AMS simulator. AMS UNL then generates the required files and adds the option `-ams_dspf_bbox` in the `xrunArgs` file to enable this feature on the simulator side.

### *Related Topics*

Using DSPF-in-the-Middle


© 2023 Cadence Design Systems, Inc.

# Using the runams Command

The `runams` command provides command-line support for netlisting the design using the AMS Unified Netlister (AMS UNL) and running simulations using xrun in batch mode.

By default, the *AMS Unified Netlisting with xrun* option is selected as the default netlisting choice when a new ADE session is opened. To enable OSS-based netlister as the default choice, specify the following in the `.cdsinit` file:

- `envSetVal( "ams.envOpts" "netlisterMode" 'string "Cellview-based")`
- `envSetVal( "ams.envOpts" "netlisterMode" 'string "OSS-based")`

If you wish to force the choice of netlister to UNL when loading an existing ADE state with OSS-based setting, specify the following:

```
envSetVal( "ams.envOpts" "forceUnlOverOSS" 'boolean t)
```

To make the netlister mode UI read only, specify the following:

```
envSetVal( "ams.envOpts" "disableNetlisterModeUI" 'boolean t)
```

`runams` will read and honor CDS environment variables to invoke the intended netlisting flow.

OCEAN also allows you to netlist and simulate from the command-line. However, `runams` provides a much simpler use model using command-line options.

**Note:** The `runams` command for AMS UNL capability is available in IC 6.1.6 ISR6, and later versions.

The `runams` command syntax is given below:

```
runams [-help | -h | -version | -V | -W | -usage]
     | runams -lib libName -cell cellName -view viewName
       action_options [setup_options] [netlisting_options] [simulation_options]
```

Use the following command to view information about all the options available in the `runams` command:

```
runams -help
```

Use the following command to view detailed information about each option available in the `runams` command:

```
runams -help optionName
```


© 2023 Cadence Design Systems, Inc.

Use the following command to view important information you must know when using runams options:

```
runams -usage
```

See the following topics for more information about the `runams` command:

- runams Command Options on page 1138
- How to Use runams Options on page 1155
- runams Command Examples on page 1157

## runams Command Options

The following table describes the `runams` command options and values.

| runams Option and Value | Description |
|---|---|
| **-h \| -help** | Displays the description of the `runams` command and its options. |
| **-h** *optionName* | Displays the detailed description for an option. |
| **-V \| -version** | Displasy `runams` version information. |
| **-W** | Displays `runams` subversion information. |
| **-usage** | Displays important information you must know when using runams options. For more information, see How to Use runams Options on page 1155. |
| **-nocdsinit** | Skips reading the `.cdsinit` file. |
| **-lib** *libName* | Specifies the library containing the configuration that you want to netlist. |
| **-cell** *cellName* | Specifies the cell containing the configuration that you want to netlist. |
| **-view** *viewName* | Specifies the cellview name of the configuration that you want to netlist. The view name can also be a cellview state. |
| **-testname** *testName* | Specifies the name of the test to be used from the given maestro view. If not specified, the first test from that maestro view is used. |


© 2023 Cadence Design Systems, Inc.

| runams Option and Value | Description |
|---|---|
| **-spectre_path** *spectreDirectoryPath* | Specifies the path to the Spectre binary. |
| **Action Options** | |
| **-netlist** | Runs the AMS UNL netlister in the specified mode. Default: `incremental` |
| `incremental` | Generates netlists for new or changed cellviews only.<br><br>This is the default netlisting mode. |
| `all` | Netlists all cellviews in the configuration, whether they have changed since the previous netlisting or not. This is equivalent to *Simulation – Netlist – Recreate* in ADE L.<br><br>**Note:** If an option effects netlisting changes (for example one of the `-netlisteropts`), use `-netlist all` to renetlist the design. |
| `skip` | Skips netlisting the design.<br><br>**Note:** Ensure that the netlist is simulatable when you select this mode. |
| **-simulate** | Runs simulation (`xrun`) in the specified mode.<br><br>Default: `batch`<br><br>**Note:** If the simulation mode has been specified through both the options, `-state` and `-simulate`, the mode specified through the `-simulate` option will be used to run the simulation. |
| `batch` | Runs xrun in the batch mode. This mode, which does not allow you to interact with the simulator, usually simulates more quickly than the `gui` mode.<br><br>This is the default simulation mode. |
| `gui` | Opens the SimVision analysis environment graphical interface that allows you to interact with the simulator using buttons, menus, and Tcl commands. |


© 2023 Cadence Design Systems, Inc.

| runams Option and Value | Description |
|---|---|
| **-savescripts** | Saves the `runSimulation` file in the *`runDir`*`/netlist` directory, but don't simulate. |
| **-clean** | Deletes all simulation snapshot files before running simulation. This will run the `xrun -clean` command to remove the `INCA_libs` directory under the *`runDir`*`/netlist` directory. |
| **-forcecheck** | Enables an in-memory check for schematic connectivity before proceeding to netlisting. |
| **-plot** | Invokes the waveform tool to plot the results in the `./`*`runDir`*`/psf` directory when simulation is completed.<br><br>The waveform tool can be specified using the `-wavetool` command.<br><br>To use the SimVision interactive debugger, run the simulation using `-simulate gui` option.<br><br>If a Tcl file containing database commands is specified using the `-tclinput` option, ensure that the database `-open` command in the file is:<br><br>`open -into ./`*`runDir`*`/psf`<br><br>If an ADE state file is used, the results will be automatically written to the `./`*`runDir`*`/psf` directory without the need to specify a separate Tcl file. If a Tcl file or a state file is not specified, `xrun` creates a directory named `abc.tran.shm` in the netlist directory, where `abc` is the name of the `tran` statement in the `analogcontrol` (scs) file. If the specified Tcl file does not have a database `-open` statement, `xrun` creates an `.shm` directory in the netlist directory, such as `ncsim.shm`. `runams` will detect this and invoke the waveform tool to plot the results in the `abc.tran.shm` or `ncsim.shm` directory. |


© 2023 Cadence Design Systems, Inc.

| runams Option and Value | Description |
|---|---|
| **Setup Options** | |
| **-cdslib** *filePath* | Specifies the `cds.lib` file to use.<br><br>If the `-cdslib` option is not specified, the standard Cadence search mechanism (CSF) is used to find the `cds.lib` file.<br><br>Relative paths are resolved with respect to the invocation directory. |
| **-file** *filePath* | Specifies a file that contains `runams` options.<br><br>Each line in the file can have one or more options. Lines in the file that are blank or begin with the comment character # are ignored. Line continuation character is not supported. For *filePath*, a relative path is resolved with respect to the invocation directory. For the options in the file, a relative path is resolved with respect to the file location. |
| **-log** *logFileName* | Writes output log messages to *logFileName*. Default: `./runams.log`<br><br>The *logFileName* that you specify with this variable interacts with the `CDS_LOG_PATH` environment variable to determine the log file name that is used. Ensure that the directory specified in the `CDS_LOG_PATH` environment variable exists.<br><br>■ If *logFileName* is an absolute path, the log file is written to *logFileName*.<br><br>■ If *logFileName* is a relative path and<br><br>❑ `CDS_LOG_PATH` is null, *logFileName* is placed in the current directory.<br><br>❑ `CDS_LOG_PATH` is non-null, the value of `CDS_LOG_PATH` is prefixed to the *logFileName*.<br><br>■ The `CDS_LOG_VERSION` environment variable also affects the final name of the log file. |


© 2023 Cadence Design Systems, Inc.

| runams Option and Value | Description |
|---|---|
| **-rundir** *runDir* | Specifies the run directory path to use.<br>Default: `./libName-cellName-viewName`<br><br>A `netlist` directory (where netlisting and simulation are run) and a `psf` directory (where the results are written) will be created under `runDir`.<br><br>Relative paths are resolved with respect to the invocation directory. |
| **-resultsdir** *resultsDir* | Specifies the results directory path to use.<br><br>By default, simulation results are written to the *runDir*`/psf` directory. If this option is specified, the simulation results will be written to the *resultsDir*`/psf` directory. If both `-resultsdir` and `-tclinput` are specified, ensure that the same results directory path is specified for *resultsDir*`/psf` and the `database -open` command in the Tcl file specified using the `-tclinput` option. |
| **-cdsenv** *filePath* | Specifies the `.cdsenv` file to use. This is the same `.cdsenv` file used by ADE.<br><br>Relative paths are resolved with respect to the invocation directory. |


© 2023 Cadence Design Systems, Inc.

| runams Option and Value | Description |
|---|---|
| **-cdsinit** *filePath* | Specifies the `.cdsinit` file to use. This is the same `.cdsinit` file used by ADE.<br><br>Relative paths are resolved with respect to the invocation directory.<br><br>You can use the `-cdsinit` option to specify a file that contains the same settings as the Virtuoso `.cdsinit` file. For example, if you have a file, `testpinchecker`, you can specify the file with this option, as shown below:<br><br>`runams -lib training -cell wx_ams -view maestro -rundir mydirsch -netlist all -testname training:wx_ams:1 -cdsinit testpinchecker`<br><br>Here, `testpinchecker` contains the following:<br><br>`aa = hnlRunPinChecker("WSM_pinCheck" "concatBusMM_tb" "config" ?checkAms t)`<br>`unless( aa`<br>`exit()`<br>`)`<br><br>In this example, the `runams` command will run the Pin Checker before the AMS UNL netlisting starts through the `runams -netlist` option. |
| **-state** | |
| *stateLoadDir*:*stateName*[:*simulatorName*] | Specifies the ADE state directory and state name to use.<br><br>The final path used to find the state file will be *stateLoadDir/libName/cellName/simulatorName/stateName*, where `libName` is the library name specified using the `-lib` option and `cellName` is the cell name specified using the `-cell` option. By default, `simulatorName` is `ams`. If `simulatorName` is specified to be something other than `ams`, the state for the specified simulator will be loaded for the ams run, and, as with ADE, the options in the state file that are not applicable to ams will be ignored.<br><br>Relative paths are resolved with respect to the invocation directory. |


© 2023 Cadence Design Systems, Inc.

| runams Option and Value | Description |
|---|---|
| *stateViewName* | Specifies the ADE cellview state to use.<br><br>The state files are found in the *libName/cellName/stateViewName* directory, where libName is the library name specified using the -lib option and cellName is the cell name specified using the -cell option. Ensure that the stateViewName is for an ams state. |
| **-switchto** | Allows to specify a different design configuration for the current maestro view.<br><br>For example, by using the following setting, you can switch the design configuration file for a test, mylib:mycell:1 from mylib.mycell:config to mylib.mycell:config2.<br><br>-switchto mylib.mycell:config2 |
| **-wavetool** | Specifies the waveform tool to use to plot the results.<br><br>Default: simvision |
| simvision | Uses the SimVision analysis environment to plot the results.<br><br>This is the default waveform tool. |
| viva | Uses the Virtuoso Visualization and Analysis tool to plot the results. |


© 2023 Cadence Design Systems, Inc.

| runams Option and Value | Description |
|---|---|
| **Netlisting Options** | |
| **-desvar** *desVars* | Specifies additional design variables, or override the value of design variables in the `cds_globals.vams` file.<br><br>The *desVars* argument is a colon-separated list of name-value pairs, such as `name1=val1:name2=val2`. During netlisting, information about design variables in the design are written to the `cds_globals.vams` file. During simulation, the `cds_globals.vams` file will be automatically sent to `xrun`.<br><br>Use the `-desvar` option to specify additional design variables you want to add in the `cds_globals.vams` file, or to specify design variables whose values you want to override in the `cds_globals.vams` file. This results in a new `cds_globals.vams` file that will be sent to xrun.<br><br>This option can be specified more than once. However, if the same design variable is specified in more than one `-desvar` statement, the value specified for that variable in the last `-desvar` statement will be used. |
| **-globalsignals** | Specifies a set of global signals, either in a file or as a list of signal names immediately following the option. |


© 2023 Cadence Design Systems, Inc.

| runams Option and Value | Description |
|---|---|
| *sigName*[=*netType*[=*netDiscipiline*[=*isGround*]]] | Specifies additional global signals, or override the declaration of design variables in the `cds_globals.vams` file.<br><br>During netlisting, information about global signals in the design are written to the `cds_globals.vams` file. During simulation, the `cds_globals.vams` file will be automatically sent to xrun.<br><br>Use the `-globalsignals` option to specify additional global signals you want to add in the `cds_globals.vams` file, or to specify global signals whose declarations you want to override in the `cds_globals.vams` file. This results in a new `cds_globals.vams` file that will be sent to xrun.<br><br>The option *sigName* is mandatory. If *netType* is not specified, `wire` will be used. If *netDiscipline* is not specified, `electrical` will be used. If `isGround` is not specified, `NO` will be used. Use single quotes to enclose the values.<br><br>Example:<br><br>`-globalsignals 'VDD50! VSS50! VDD30!=wire=electrical=NO VSS30!=wire=electrical=YES'.`<br><br>This option can be specified more than once. However, if the same global signal is specified in more than one `-globalsignals` statement, the arguments specified for that signal in the last `-globalsignals` statement will be used. |


© 2023 Cadence Design Systems, Inc.

| runams Option and Value | Description |
|---|---|
| *sigFile* | Specifies the file containing global signal names, each name on its own line in the file.<br><br>Each line in the file must be in the format: *sigName*[=*netType*[=*netDiscipiline*[=*isGround*]]].<br><br>Use single quotes to enclose the values. The signals will be placed in the `cds_globals.vams` file that was created during netlisting, and the resulting new `cds_globals.vams` file will be sent to xrun.<br><br>Example:<br><br>`'VDD50! VSS50! VDD30!=wire=electrical=NO VSS30!=wire=electrical=YES'.` |


© 2023 Cadence Design Systems, Inc.

| runams Option and Value | Description |
|---|---|
| **-netlisteropts** *'option1:option2'* | Specifies AMS UNL options. The options argument is a colon-separated list of name-value pairs enclosed in single quotes, such as:<br>`-netlisteropts 'amsPortConnectionByNameOrOrder=name:useSpectreInfo=spectre veriloga spice'.`<br>The following netlister options can be set using the `-netlisteropts` command:<br>■ `amsPortConnectionByNameOrOrder=name order`<br>Specify how to print the port connection. Default: `name`).<br>■ `useSpectreInfo`<br>Specify the list of views that are to be netlisted as spectre. The default is "`spectre veriloga spice`". You may need to add `symbol` to this list if you want symbol views to be netlisted as spectre.<br>This option can be specified more than once. However, if different values are specified for an argument in more than one `-netlisteropts` statement, the value specified for that argument in the last `-netlisteropts` statement will be used. |
| **-portdrill** | Enables the port drilling feature for *runams*. Valid values for this option are `all`, `except_top`, and `no`.<br>Default: `all`. |
| **Simulation Options** | |
| **-solver** | Specifies the analog solver to use.<br>Default: `spectre` |
| `spectre` | Specifies that the Spectre solver is to be used.<br>This is the default option. |


© 2023 Cadence Design Systems, Inc.

| runams Option and Value | Description |
|---|---|
| `ultrasim` | Specifies that the UltraSim solver is to be used. |
| **-analogsolver** | Lets you specify a host for Spectre.<br><br>Examples:<br><br>Runs Spectre on the local host. You can also use a script or a resource scheduler, provided it boots Spectre.<br><br>`runams -analogsolver "spectre"`<br><br>Runs Spectre on a remote host.<br><br>`runams -analogsolver "bsub -R \"OSNAME==Linux &7 OSREL==EE70 && SFIPLATFORM==c0700\" -n 1 -q lnx64 -J spectre"` |
| **-amscontrol** *filePath* | Specifies the ams control file to use. Typically this file contains the ams control block. The *filePath* argument is passed to the simulator. |
| **-analogcontrol** *filePath* | Specifies the analog simulation control file to be used. This file contains the spectre, ultrasim or aps control statements, such as the `tran`, `info` or `options` statements.<br><br>The *filePath* argument is passed to the simulator. |
| **-cdsglobals** *fileName* | Specifies a file that includes the `cds_globals` module definition.<br><br>The `cds_globals` module declares design variables and global signals. The specified *fileName* will be sent to xrun.<br><br>**Note:** The `cds_globals.vams` file that is created during netlisting will not be sent to xrun. The *fileName* argument is passed to the simulator. |
| **-connectrules** | Specifies the built-in and user-defined connect rules to use.<br><br>This option can be specified more than once. |


© 2023 Cadence Design Systems, Inc.

| runams Option and Value | Description |
|---|---|
| [*libName*.]*ruleName*[:*viewName*] | Specifies the built-in connect rule to use in the `library.cell:view` format, such as:<br>`-connectrules connectLib.connRule_5v_full:connect` |
| `userDef:`[*ruleName*][:*fileName*] | Specifies the user-defined connect rule to use in the `rulename:filename` format, such as:<br>`-connectrules userDef:connRule_5v_full:/path/file1`<br>The *ruleName* and *fileName* are passed to the simulator.<br>This option can be specified more than once. |
| **-iecard** *filePath* | Specifies the path to an `ie_card.scs` file that contains information about how the connect rules are used during simulation. |
| **-discipline** *disciplineName* | Specifies the default discipline for digital nets for which a discipline is either not specified or cannot be determined through discipline resolution.<br>If `-discipline` is not specified, `-discipline logic` will be printed in the `runSimulation` file. Use `-discipline none` to disable printing of `-discipline logic` in the `runSimulation` file. |
| **-f** *filePath* | Specifies the files containing xrun command-line options to be passed as the argument to the `xrun -f` command. The *filePath* argument is a colon-separated list of file paths, such as:<br>`-f filePath1:filePath2`<br>The *filePath* argument is passed to the simulator.<br>This option can be specified more than once. |
| **-hdlvar** *filePath* | Specifies the name of the `hdl.var` file to use.<br>The *filePath* argument is passed to the simulator. |


© 2023 Cadence Design Systems, Inc.

| runams Option and Value | Description |
|---|---|
| **-incdir** *path* | Specifies the directory to search for `include files. The *path* argument is a colon-separated list of directory paths, such as:<br>`-incdir path1:path2`<br>The path argument is passed to the simulator.<br>This option can be specified more than once. |
| **-irunopts** *options* | Specifies additional xrun command-line options, such as:<br>`-irunopts '-iereport +dr_info'`<br>Use single quotes to enclose the command-line options. These command-line options are passed as is to xrun.<br>xrun is run in the *runDir*/netlist directory. Therefore, all paths must be absolute or relative to the netlist directory. |
| **-modelfile** *modelFiles* | Specifies the analog model files and section name. The *modelFiles* argument is a colon-separated list of files with model section names enclosed in parentheses, such as:<br>`-modelfile 'file1:file2(NN)'`<br>Use single quotes to enclose the value when section names of a model are specified. The *modelFiles* argument is passed to the simulator.<br>This option can be specified more than once. |
| **-path** *path* | Specifies the model include directories setup. This is equivalent to the `xrun -modelincdir` option. The *path* argument is a colon-separated list of directory paths, such as:<br>`-path path1:path2`<br>The *path* argument is passed to the simulator.<br>This option can be specified more than once. |


© 2023 Cadence Design Systems, Inc.

| runams Option and Value | Description |
|---|---|
| **-reportinvalidbinding** | Stops the simulation with an error message when an invalid binding is found in an HED view. This option provides the same functionality as the environment variable `AMS_UNL_INVALID_BINDING_ACTION`. |
| **-tclinput** *filePath* | Specifies a Tcl file to use.<br><br>The Tcl file normally contains the `database -open` command and the `probe` statements that specify nets and terminals to save for plotting, although any Tcl command that xrun accepts can be specified in the Tcl file.<br><br>The `database -open` command in the Tcl file must open the results database in `./`*`runDir`*`/psf`, as this is the directory that will be passed to the waveform tool by the `-plot` option. The *`filePath`* argument is passed to the simulator. |
| **-v** *filePath* | Specifies the Verilog files to be passed as the argument to the `xrun -v` command.<br><br>The *`filePath`* argument is a colon-separated list of file paths, such as:<br><br>`-v filePath1:filePath2`<br><br>The *`filePath`* argument is passed to the simulator.<br><br>This option can be specified more than once. |
| **-y** *path* | Specifies the directories to be passed as the argument to the `xrun -y` command.<br><br>The *`path`* argument is a colon-separated list of directory paths, such as:<br><br>`-y path1:path2`<br><br>The *`path`* argument is passed to the simulator.<br><br>This option can be specified more than once. |

**IP Export Options**


© 2023 Cadence Design Systems, Inc.

| runams Option and Value | Description |
|---|---|
| **-export** | Enables the Virtuoso IP export reuse flow. The generated export files are placed in the directory specified through the `-rundir` option |
| **- testbenchtopcell** *topCellName* | Specifies the design library and cell names to be used for System Verilog Configuration File generation. |
| **- iplabel** *labelName* | Specifies the label name to be used for `.amsbind.scs` file generation. |
| **- v2001configname** *configName* | Specifies the configuration name to be used for System Verilog Configuration File generation |
| **-ipencrypt** | Enables encryption for the exported IP. Valid values for this option are `yes`, `no`. |
| **-iplabelAddOnLib** | Adds the specified IP label name as the prefix to the Virtuoso library name. |
| **-iphierpathtoiptopcell** | Specifies the hierarchy path from the top cell to the current IP. |
| **-ipselectsimopt** | Specifies the simulation options to be printed in the argument file `xrunArgs`. |
| *all* | Prints all the simulation options to the argument file. |
| *modelsetup* | Prints the model file settings to the argument file. |
| *perfsetup* | Prints the simulation options from the *High Performance Simulation Options* form. |
| *iesetup* | Prints the IE Card settings from the *Interface Element (IE) Setup* form. |
| *analogctrlfile* | Prints the analog control files.<br>For example, `amsControlSpectre.scs` |

Note the following:

- By default, environment variables specified in file paths are expanded in the `runSimulation` file created by the `runams` command. If you want the `runSimulation` file created by `runams` to be portable, use single quotes around file paths so that the environment variables in the file paths are not expanded in the `runSimulation` file.


© 2023 Cadence Design Systems, Inc.

**Note:** Relative paths specified using the ~ or . characters will be resolved with respect to the invocation directory (the directory in which you ran the `runams` command).

- The following `amsdesigner` command options are supported by `runams` for compatibility purposes. However, Cadence recommends using the equivalent `runams` options given below.

| amsdesigner Command Options Supported by runams | Equivalent runams Command Options |
|---|---|
| **-ncvlogopts**<br>**-ncvhdlopts**<br>**-ncelabopts**<br>**-ncsimopts** | **-irunopts** |
| **-modelpath** | **-modelfile** |
| **-modelincdir** | **-path** |
| **-input** | **-tclinput** |


© 2023 Cadence Design Systems, Inc.

## How to Use runams Options

This section describes important information you must know when using runams options.

1. If `-help`,`-version`, `-V`, `-W` or `-usage` is specified with other options, those options will be ignored.
2. To run simulation, you must specify the `-cdsenv`, `-state` or `-analogcontrol` option, or use a cellview state with the `-view` option.
3. If `-clean` is specified without `-simulate` or `-savescripts`, the `xrun -clean` command is run to remove the INCA_libs directory in the netlist directory.
4. Only the following options can be specified more than once. If other options are specified more than once, the value specified for the last instance of those options will be used.

   ```
   -file
   -desvar
   -globalsignals
   -netlisteropts
   -connectrules
   -f
   -incdir
   -path
   -modelfile
   -irunopts
   -v
   -y
   ```

   For `-desvar`, if the same design variable is specified in more than one `-desvar` statement, the value specified for that variable in the last `-desvar` statement will be used.

   For `-globalsignals`, if the same global signal is specified in more than one `-globalsignals` statement, the arguments specified for that signal in the last `-globalsignals` statement will be used.

   For `-netlisteropts`, if different values are specified for an argument in more than one `-netlisteropts` statement, the value specified for that argument in the last `-netlisteropts` statement will be used.

   If `-connectrules`, `-f`, `-incdir`, `-path`, `-modelfile`, `-v` and `-y` are specified more than once, the values specified for all instances of the options be used.
5. If `-file` option is specified, the options in the file will be appended after other command-line options. Then the rules specified in point 4 above are used to process all the options. For example, the `cmdFile` file includes:


© 2023 Cadence Design Systems, Inc.

```
"-view config -path path2 -desvar var1=2:var2=3
```

and you specify:

```
runams -lib lib -cell cell -view view -netlist -desvar var1=1 -
path path1 -file cmdFile"
```

This is like using:

```
runams -lib lib -cell cell -view view -netlist -desvar var1=1 -path path1 -view
config -path path2 -desvar var1=2:var2=3
```

As per the rules in point 4 above, these commands will be applied as:

```
runams -lib lib -cell cell -view config -netlist -path path1:path2 -desvar
var1=2:var2=3
```

6. If the `-lib`, `-cell` and `-view` options point to a cellview state, runams will first get *`topLib.topCell:topView`* from the cellview state, then run `runams -lib` *`topLib`* `-cell` *`topCell`* `-view topView -state` *`state_view_name`*. If the `-state` option is also specified, runams will ignore the `-state` option.

7. If an option is specified in both the `.cdsenv` file and an ADE state, the option in the ADE state has precedence. However, any option specified in the `.cdsenv`, that does not have a GUI /state equivalent, will be used from the `.cdsenv` file.

8. If `-state`, `-cdsenv`, and `-analogcontrol` or `-tclinput` are specified, runams reads the `.cdsenv` file first and reads the state files next. The files specified using the `-analogcontrol` or `-tclinput` options will be used instead of the files that would have been created by the state file.

9. If you specify the `-cdsenv` and the `-analogcontrol` options, but do not specify the `-state` option, the file specified using `-analogcontrol` option is used. runams will not use the file specified using the `-cdsenv` option to generate the analog control, Tcl or `runSimulation` file, but the settings that affect netlisting in the `.cdsenv` file will take effect during netlisting.

10. If conflicts occur between the options specified by `-irunopts`, command-line options and the options in the `.cdsenv` or a state file, the options specified by `-irunopts` take precedence, then command-line options, then the options in the `.cdsenv` or a state file. For example, if there is a conflict in the values specified for the `-solver` option in the command-line and in the `.cdsenv` or a state file, the values specified in the command-line options will be used.


© 2023 Cadence Design Systems, Inc.

## runams Command Examples

- Renetlists the entire design and runs a simulation.

```
runams -lib mylib -cell top -view config -netlist all -simulate
```

- Netlists only the cellviews in the design that were changed since the previous netlisting, but does not run a simulation (because `-simulate` is not specified).

```
runams -lib amsLib -cell top -view config -netlist incremental
```

- Specifies a run directory `myrundir` where the netlist and data files will be placed.

```
runams -lib mylib -cell top -view config_ams -netlist all -simulate -rundir
myrundir
```

- Specifies that simulation be done in the interactive mode using SimVision analysis environment graphical interface. This command also specifies a model file.

```
runams -lib mylib -cell top -view config_ams -netlist all -simulate gui -rundir
myrundir -modelfile myModels.scs:fastModels.scs(ff)
```

- Sets the AMS analog solver to `ultrasim`.

```
runams -lib mylib -cell top -view config_ams -netlist all -simulate -rundir
existingrundir -solver ultrasim
```

- Netlists the configuration `mylib.top:config` and saves the `runSimulation` file (using `-savescripts`) without simulating (because `-simulate` is not specified).

```
runams -lib mylib -cell top -view config -netlist -savescripts -analogcontrol
$CDIR/analog.scs -tclinput probe.tcl -path $CDIR -modelfile
myModels.scs:fastModels.scs(ff) -rundir top_run1 -connectrules
ConnRules_5V_full
```

- Uses the setup in the `state_ams` ADE state file to netlist and simulate a configuration.

```
runams -lib amsPLL -cell pll_160MHZ_sim -view config -netlist -simulate -rundir
run1 -state artist_states:state_ams -log ./logs/myRun.log
```

- Uses the setup in the `ams_state1` ADE cellview state to netlist and simulate a configuration.

```
runams -lib amsPLL -cell pll_160MHZ_sim -view config -netlist -simulate -state
ams_state1
```

- Uses the setup in an analog simulation control file named `amsControl.scs` to netlist and simulate a configuration, and specifies a model file, a user defined connect rule, and a TCL file. You can use environment variables (such as `MODEL_DIR` used in this example) when specifying values for options.

```
runams -lib testLib -cell top -view config -netlist -simulate -rundir run2 -
analogcontrol './setup/amsControl.scs' -modelfile $MODEL_DIR/models/spectre/
gpdk090.scs'(NN)' -cdsglobals './setup/cds_globals.vams' -connectrules
userDef:ConnRules_25V_full_fast:connectLib/ConnRules_25V_full_fast.vams -
tclinput './setup/probe.tcl' -log './logs/run_control.log'
```

- Uses the specified maestro view to netlist and export the design configuration into a portable run directory using the IP export option `-export`.


© 2023 Cadence Design Systems, Inc.

While setting this command, it is assumed that the Virtuoso IP would be instantiated into a larger digital testbench, as specified by the digital testbench and Verilog config scope export settings defined in the Virtuoso ADE forms.

```
runams -lib zambezi45_sim -cell pll_sim -view maestro -netlist all -clean -
cdslib /home/user1/cds.lib -rundir /home/user1/runams_export_rundir -export -
log runams_export.log
```

**Note:** This command must be set after specifying the values for the settings in the AMS IP Export form that are specific to the `-export` option.

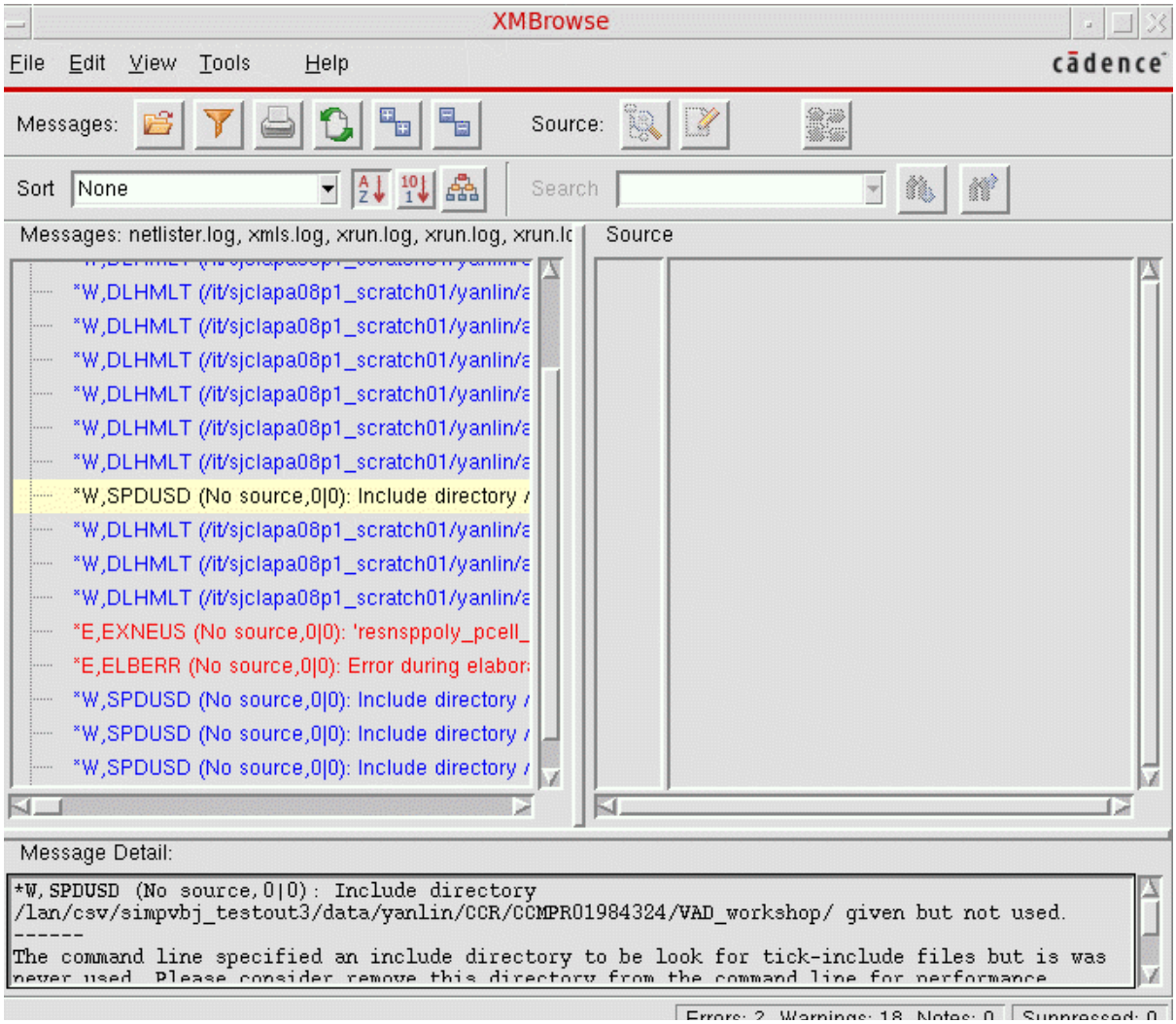
- Uses the configuration view `config_wreal` to create a netlist and exports the design configuration into a portable run directory using the IP export option `-export`. While setting this command, it is assumed that the IP will be instantiated into the larger digital testbench `tb_dut` at the hierarchical instance path `tb_dut.I1.I2` and have all the bindings defined by a top level Verilog configuration called `myTopConfig`.

  ```
  runams -lib zambezi45_sim -cell pll_sim -view config_wreal -netlist all -clean
  -cdslib /home/user1/cds.lib -rundir /home/user1/runams_export_rundir -export -
  ipselectsimopt all -testbenchtopcell tb_dut -iphierpathtoiptopcell
  tb_dut.I1.I2 -iplabel config_function -v2001configname myTopConfig -log
  runams_export.log
  ```


© 2023 Cadence Design Systems, Inc.

## Viewing the Output Log for AMS

You can view the output logs (netlister.log and xrun.log) from the *Simulation – Output Log* submenu. Choose *Output Log – LogFile Utility* to launch the XMBrowse message browser.



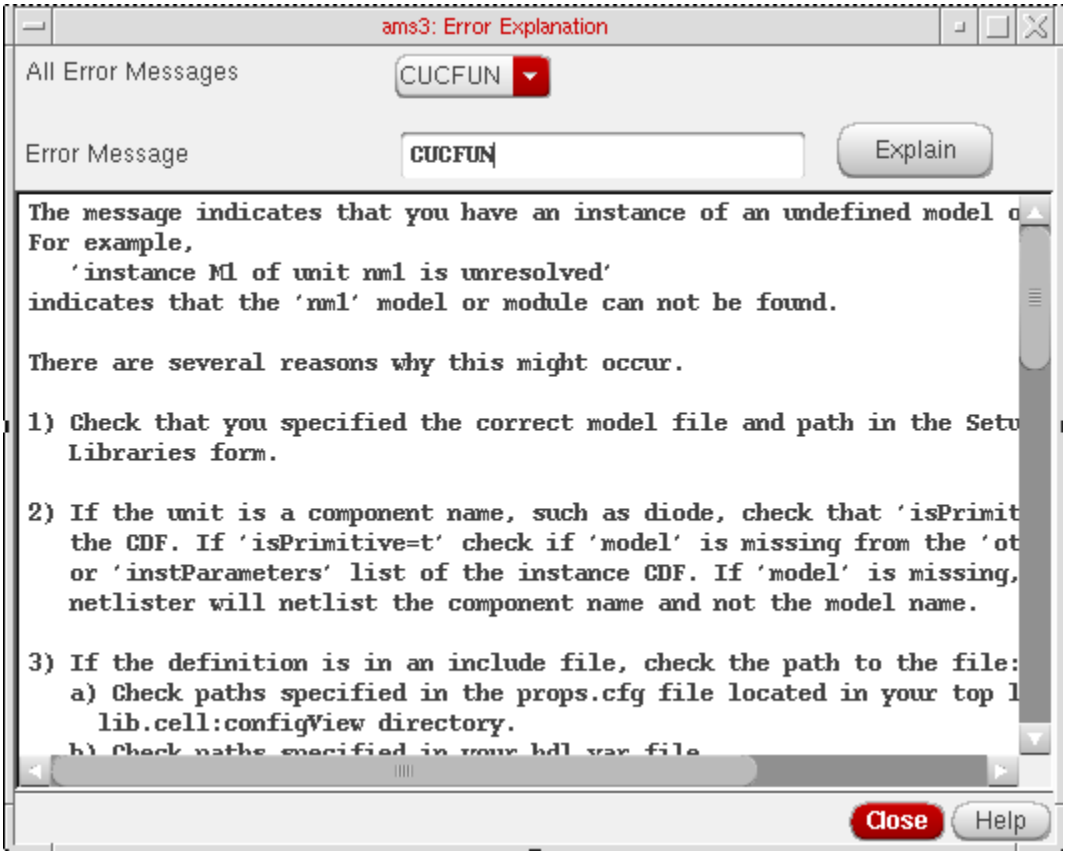

You can analyze log files with the XMBrowse utility. The XMBrowse utility lets you select a log file message and view the source code that caused the message. It includes sorting and filtering tools that let you view only those messages that are important to you. You can also use the message browser to print formatted output reports. To know more about the XMBrowse utility, refer to the *XM Browse Message Browser User Guide.*

Similarly, choose the *Netlister Log*, *Compiler Log*, *Elaborator Log*, *Simulator Log, 3-step Log* or *xrun Log* options to view the respective logs in the *Results Browser*.


© 2023 Cadence Design Systems, Inc.

## Viewing the Error Explanation for AMS

You can view detailed explanation of the error for AMS in the Error Explanation form. Choose *Simulation – Output Log – Error Explanation* to launch the Error Explanation form.




All the error messages that are present in the log files that are created in the psf directory while a session is being run are listed in the All Error Messages field. To view details about an error message, you can select the Error from All Error Messages field and then click the Explain button. The description for the error appears as shown in the figure above.

You can also enter the error in the Error Message field. If you enter a string, it will be treated as an error string. If you enter two strings, the first string is treated as the tool name and the second string is treated as an error message. If you enter more than two strings, you will get a warning.

The error message description consists of the following:

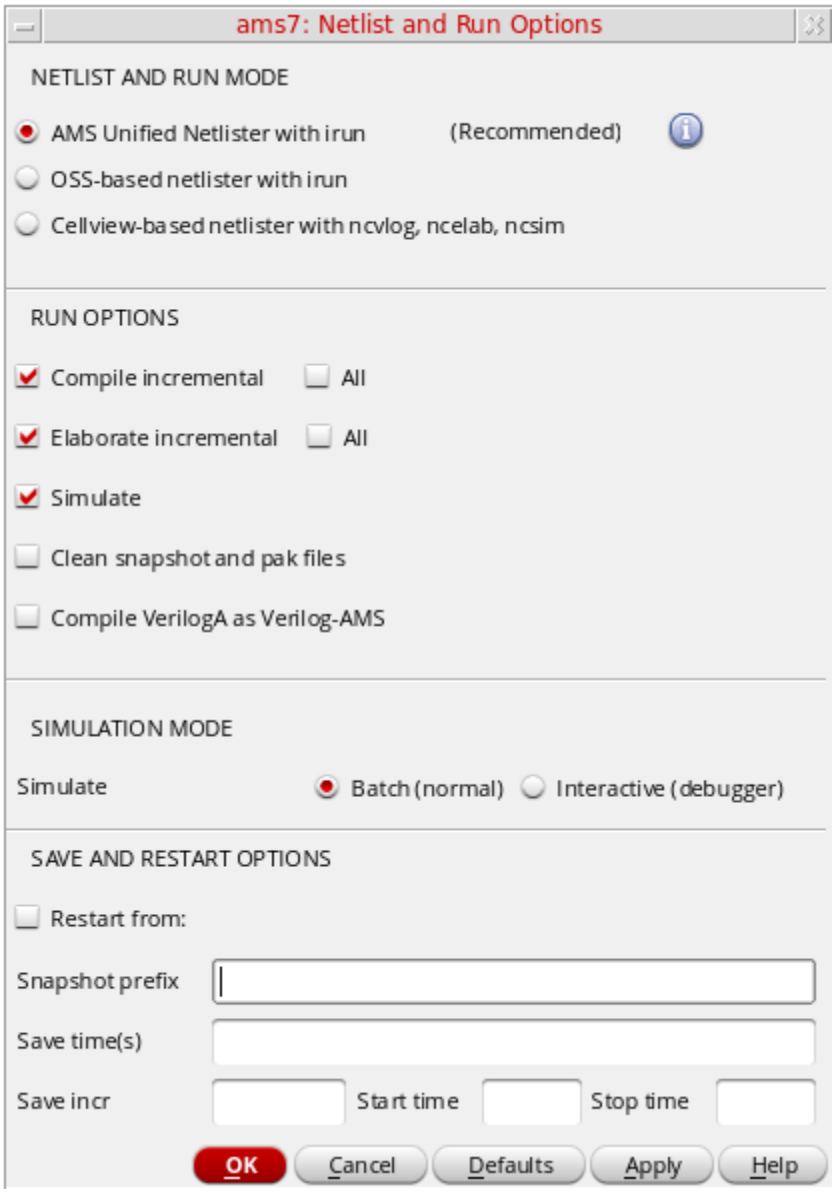
- For a growing number of messages, special messages are available to describe the problem and if known, a solution for the same. The number of messages for which there will be additional help is expected to grow with each release.


© 2023 Cadence Design Systems, Inc.

For all error names, the xmhelp for that error name is printed.

## Using the SimVision Debugger

You can set AMS netlister and simulation run options by choosing *Simulation – Netlist and Run Options*. This brings up the *Netlister And Run Options* form.




You can choose to only compile, elaborate or simulate the design, although by default all three are selected as shown in the screenshot. For example, if you select only *Elaborate*, the *Simulation – Netlist* command only elaborates the design. Compilation and elaboration can


© 2023 Cadence Design Systems, Inc.

be incremental or for the whole design together. A simulation would fail if you choose both *Compile* and *Simulate*. You would need to either deselect *Simulate* or select *Elaborate* as well.

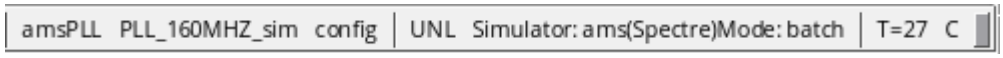
The *Batch* mode is the default mode. In this mode, all signal plotting occurs in the default analog waveform plotting tool– Virtuoso Visualization and Analysis XL.

The *Interactive* mode launches the complete SimVision debug environment interactively on top of the current design. SimVision is a graphical user interface for Cadence digital and mixed-signal simulators and related full digital debugging tool suite.

Direct Plot and plotting from the Calculator or browser go to the selected waveform tool, regardless of the run mode or whether or not SimVision is up.

ADE Explorer sends Tcl commands to SimVision for the simulation run and to plot the signals listed in the outputs pane. This mode lets you control xmsim and simvision from the console window.

The ADE Explorer window displays the selected mode on the status bar as highlighted in this snapshot.



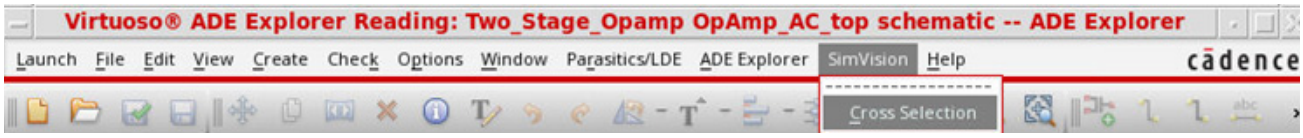

Note the following:

- Currently, the AMS simulator does not support multi-point and corner-enabled simulation runs in the *Interactive (debugger)* mode.
- To use the *Interactive (debugger)* mode for a single point or corner, ensure that the *Optimize Single Point Run* check box in the *Job Policy Setup* form is selected.

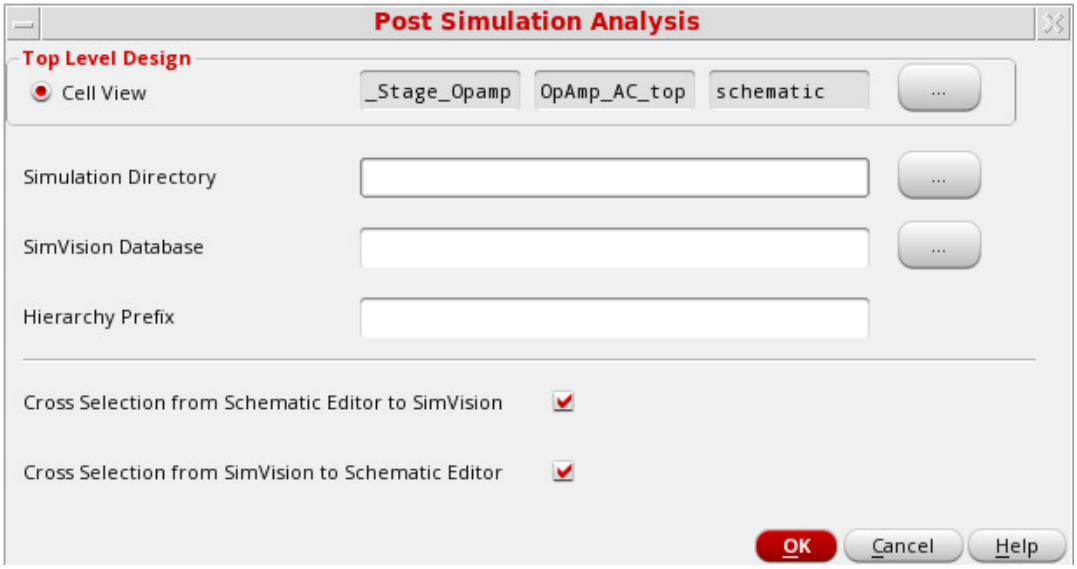

© 2023 Cadence Design Systems, Inc.

## Setting Up a Cross Selection Menu to Launch the Post Simulation Analysis Form

By default, the *Virtuoso Schematic Editor* does not have a SimVision menu. To add it to the toolbar, select *Launch – Plugins – Mixed Signal Options – SimVision*.




In the *SimVision* menu, click *Cross Selection*. The *Post Simulation Analysis* form is displayed.




You can specify the following information in the form. Click the browse button next to a field to select the input data:

| Field | Description |
|---|---|
| *Cell View* | Specify the top-level design cellview. |
| *Simulation Directory* | Specify the path of the simulation database. |
| *SimVision Database* | Specify the path and filename of the SimVision database `.trn` file. |


© 2023 Cadence Design Systems, Inc.

| Field | Description |
|---|---|
| *Hierarchy Prefix* | This field includes the hierarchy prefix used in the testbench, such as `test.top` or `test:top`.<br><br>**Note:** You can use a period (.) or a colon (`:`) as the delimiter in the hierarchy prefix. |

Additionally, you can use the following check boxes to migrate from the Virtuoso Schematic Editor to the Virtuoso Verilog Environment for NC-Verilog and vica-versa.

- *Cross Selection from Schematic Editor to SimVision:* select this check box to migrate from the *Virtuoso Schematic Editor* to the *Virtuoso Verilog Environment for NC-Verilog*.
- *Cross Selection from SimVision to Schematic Editor:* select this check box to migrate from the *Virtuoso Verilog Environment for NC-Verilog* to the *Virtuoso Schematic Editor*.

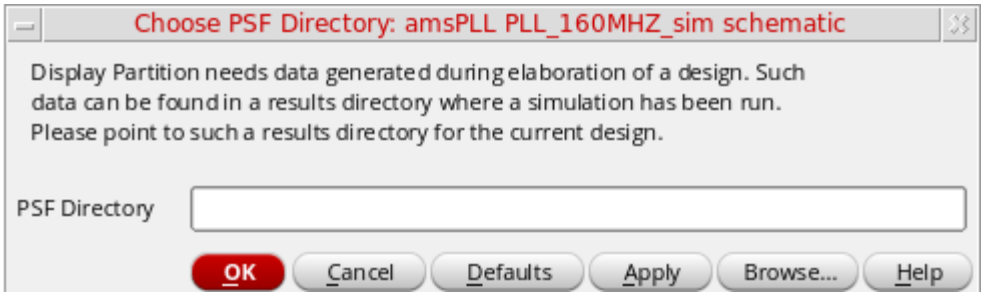

© 2023 Cadence Design Systems, Inc.

## Display Partition

Once you elaborate a design or run a simulation, you can view your data and distinguish between analog instances or nets, digital/real instances or nets, and mixed instances or nets by the color associated with each partition.

1. To access this feature, choose *Launch – Plugins – Mixed Signal Options – AMS* in your schematic window. The *AMS* option appears on the menu bar.
2. Choose *AMS – Display Partition – Initialize*. If you choose the Initialize menu item after running a simulation in ADE Explorer, the *Display Partition* menu items are enabled.

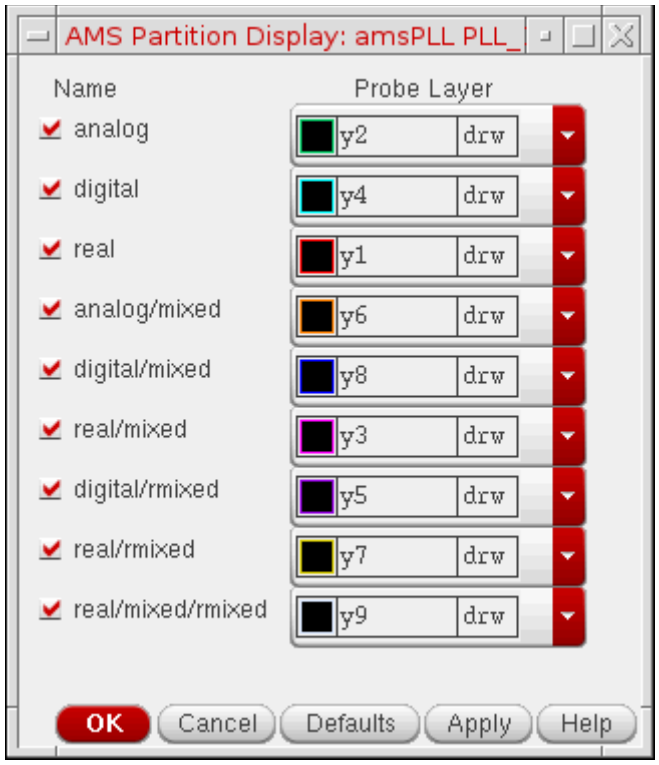
   However, if you do not run the simulation, the *Choose PSF Directory* form is displayed, as shown below. On this form, type the path to the simulation data directory in the *PSF Directory* field, or select the results using the *Browse* button.





© 2023 Cadence Design Systems, Inc.

3. Choose *AMS – Display Partition – Interactive* to view the *Partition Display* window.




This feature highlights the analog, digital, and real parts of a schematic in different colors.

**analog** indicates that the net or instance is analog in nature. A net or instance is analog throughout a hierarchy if everything under that instance is analog.

**digital** indicates that the net or instance is digital in nature. A net or instance is digital throughout a hierarchy if everything under that instance is digital.

**real** indicates that the net or instance is wreal in nature. A net or instance is real throughout a hierarchy if everything under that instance is real.

**analog/mixed** indicates that the net is mixed in nature. It means that some segments of the net or instance are analog and some are digital in the hierarchy.

**digital/mixed** indicates the same as analog/mixed, the difference being that at the current level, the net is digital.

**real/mixed** indicates the net is wreal and connecting to R2E, or the instance has both wreal and electrical inside.

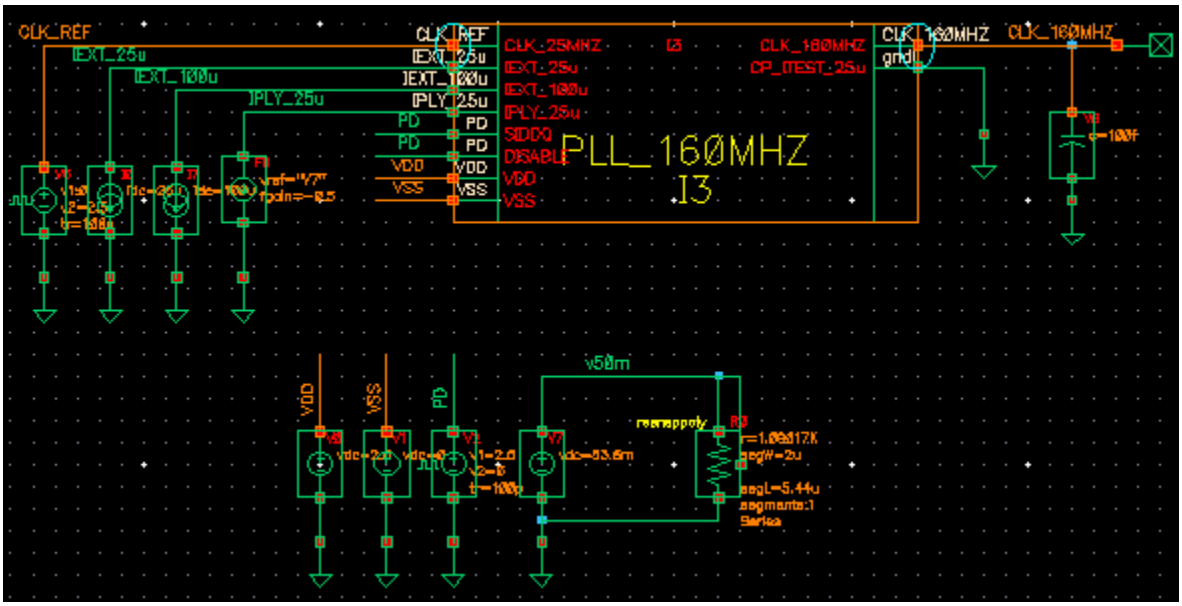
**digital/rmixed** indicates the net is a logic net connecting to R2L, and R2L is connected to wreal net.


© 2023 Cadence Design Systems, Inc.

**real/rmixed** indicates the net is wreal net connecting to R2L, or the instance has both wreal and logic inside.

**real/mixed/rmixed** indicates the net is wreal net connecting to both R2L and R2E, or the instance has wreal,electrical and logic nets inside.

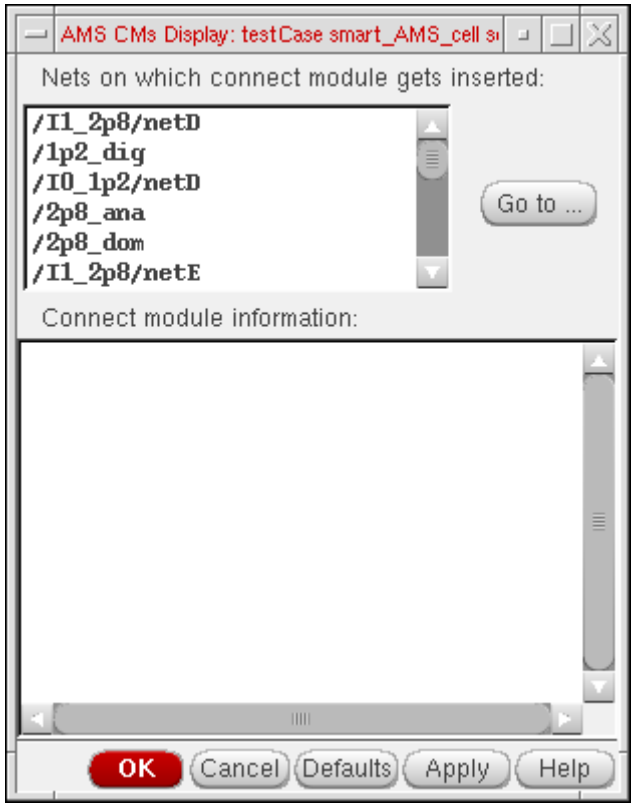
The schematic reflects these color preferences.





© 2023 Cadence Design Systems, Inc.

4. You can see all the IEs in the configuration by choosing *AMS – Display Partition – IE Information*. This brings up the *AMS CMs Display* dialog box displaying all IEs.




You can select an IE and click *Go to* to see it zoomed-in on the schematic.


© 2023 Cadence Design Systems, Inc.

## Default Digital Discipline Selection

Disciplines denote an object's domain scope as being electrical or analog (with an electrical discipline, for example) or digital (with a logic discipline, for example). You can define the default disciplines for design objects by using the *AMS – Default Digital Discipline Selection* command in the Composer window.

**Note:** This option affects the settings only on the CR/CM setup form and not the IE card setup form.

Default digital discipline selection indirectly controls the selection of a connect module (IE) on mixed domain nets. A discipline denotes an object as analog or digital based on whether it is assigned the electrical or logic type discipline, respectively. When objects of different domains/disciplines are connected, connect rules help determine which connect modules are inserted between the mixed nets/ports.

The inserted connect modules then convert signals to values that are appropriate for each discipline. To customize the conversions for your design, you can use the connect rules to override parameters, such as supply voltage or rise time, that are used in the connect modules.

For more information, see the *Mixed-Signal Aspects of Verilog-AMS* chapter of the *Cadence Verilog-AMS Language Reference*.

***To specify a default digital discipline for design objects,***

1. Choose *Launch – Plugin – Mixed Signal Options – AMS* in your schematic window.

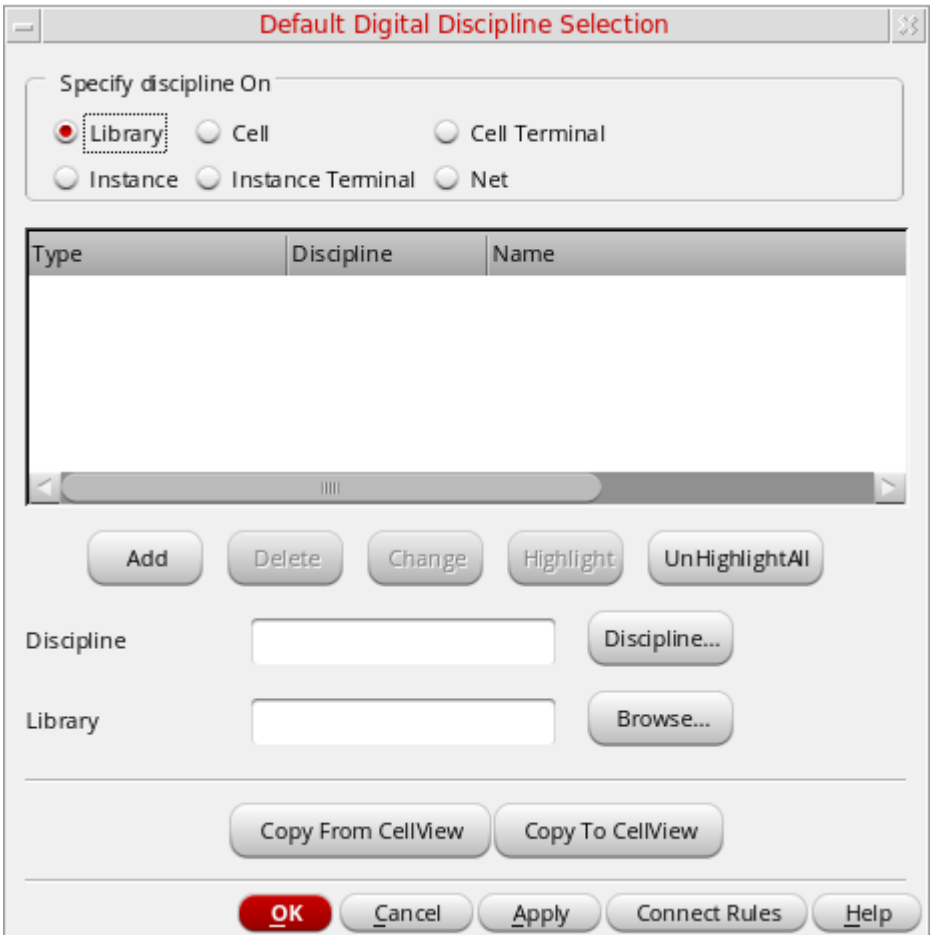
   The *AMS* menu appears on the menu bar.

2. Choose *AMS – Default Digital Discipline Selection*.

   A submenu appears showing six options: *Library*, *Cell*, *Cell Terminal*, *Net*, *Instance*, *Instance Terminal*.


© 2023 Cadence Design Systems, Inc.

3. Select any of the submenu options to bring up the Default Digital Discipline Selection form. For example, if you select *Library*, the form comes up as shown here. You can specify disciplines on libraries in this form.




Alternatively, you can select any of the other options from the *Specify discipline on* group box. The fields below the *Discipline* field change as follows depending on the design object selected.

| Design Object | Related fields | How to specify |
|---|---|---|
| *Library* | *Library* | Type in a valid value or use the *Browse* button to specify a library from the Library Browser. |
| *Cell* | *Library*<br>*Cell* | Type in valid values or use the *Browse* button to specify a library and cell from the Library Browser. |


© 2023 Cadence Design Systems, Inc.

| Design Object | Related fields | How to specify |
|---|---|---|
| *Cell Terminal* | *Library*<br>*Cell*<br>*Terminal* | Type in valid values or use the *Select* button to select a cell terminal from the schematic. |
| *Net* | *Net* | Type in a valid value or use the *Select* button to select a net from the schematic. |
| *Instance* | *Instance* | Type in a valid value or use the *Select* button to select an instance from the schematic. |
| *Instance Terminal* | *Instance Terminal* | Type in a valid value or use the *Select* button to select an instance terminal from the schematic. |

4. The table lists the types of design objects, their default disciplines, and their names. You can sort this table by *Type* or *Discipline*.
5. To create a new discipline,
   a. Click the *Discipline* button.

      The Create Discrete Disciplines form appears in which you can create a discipline. The new discipline appears in the *Discipline* field.

   b. Click the *Add* button.


© 2023 Cadence Design Systems, Inc.

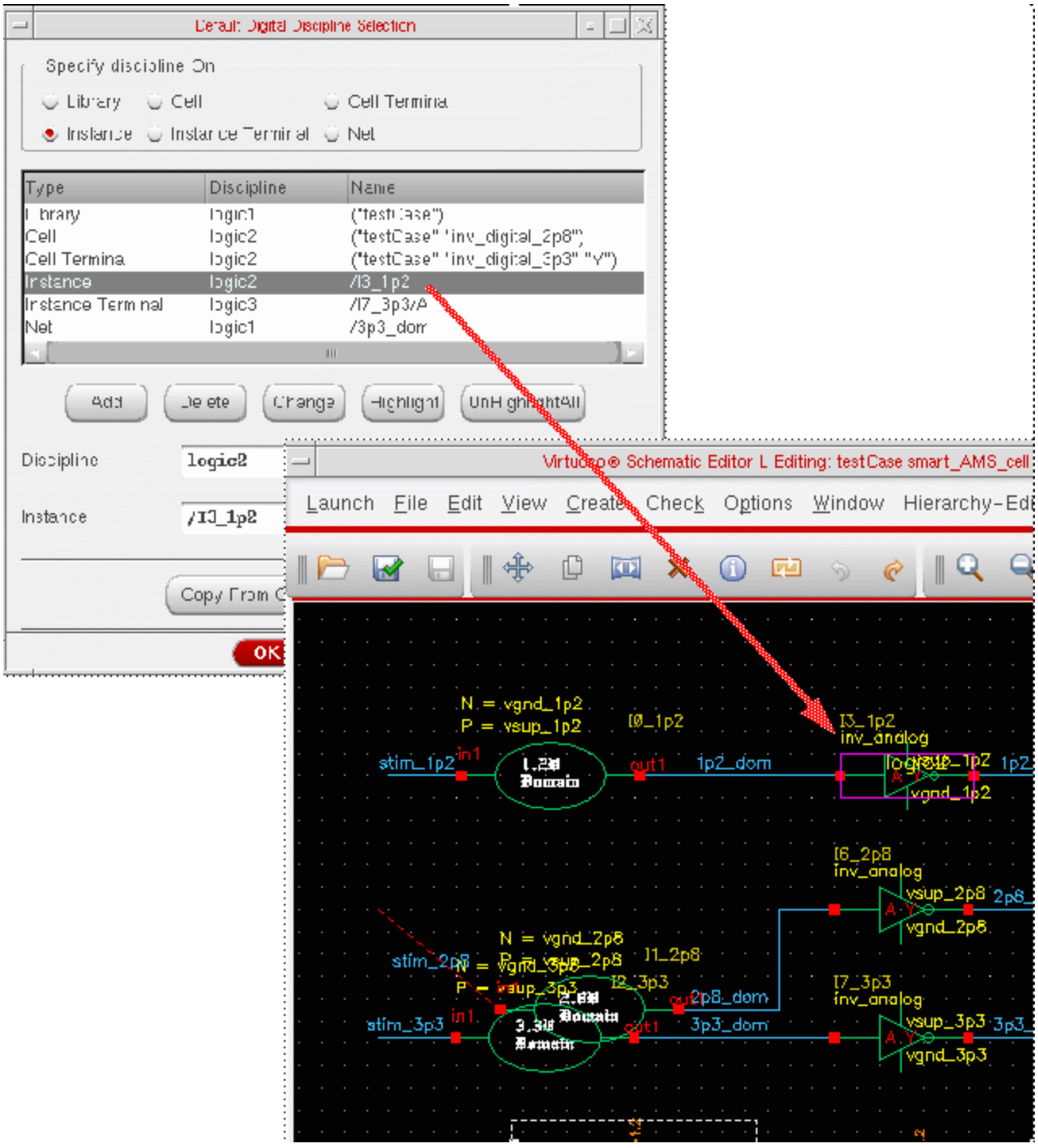
6. To delete one or more disciplines specified on an object,
   a. Select one or more rows in the table.
   b. Click the *Delete* button.
7. To change a discipline specified on an object,
   a. Select a row in the table.

      The form refreshes to show the fields pertaining to the selected kind of object.
   b. Change the values as required and click the *Change* button.
8. To view a discipline on the schematic,
   a. Select a row from the table.

      Only nets, instances and instance terminals can be highlighted.
   b. Click the *Highlight* button.


© 2023 Cadence Design Systems, Inc.

As shown in the illustration below, the selected discipline, in this case `logic_1p2`, appears highlighted in the schematic. You can select and highlight multiple objects this way. You can click the *Unhighlight All* button to remove the highlights.





© 2023 Cadence Design Systems, Inc.

9. To copy disciplines from the cellview, click the *Copy from Cellview* button. Conversely, to copy disciplines from the form to the cellview, click the *Copy to Cellview* button.

   After copying over disciplines from a cellview, when you click the *Discipline* button, a form may pop up listing the disciplines that have not been defined and asking if you would like to define them. If you select *Yes*, the Create Discrete Disciplines form appears showing a list of the undefined disciplines. If you select *No*, it appears blank.

10. You can specify connect rules for a selected discipline by using the *Connect Rules* button to bring up the Select Connect Rules form. You can use this form to create, modify or delete connect rules. This form appears only if ADE Explorer is up and the simulator set to ams. Otherwise, an error message appears.

11. Click *OK*.

    The disciplines created are automatically compiled.


© 2023 Cadence Design Systems, Inc.

# 16

# Working with HSPICE Simulator

This chapter decibels the interface used to specify the HSPICE simulator in Virtuoso ADE Explorer and how to specify the HSPICE simulator options.


- Introduction to HSPICE Direct Simulator on page 1176
- HspiceD Features on page 1179
- Specifying HSPICE Options on page 1181
- Setting up Reliability Option on page 1187



© 2023 Cadence Design Systems, Inc.

# Introduction to HSPICE Direct Simulator

The *Virtuoso ADE Explorer* contains a direct integration of the *HSPICE* simulator. ADE's *HSPICE* integration (*HspiceS*) used to be based on the socket methodology, which required edit permissions to the schematic being simulated, and caused usage issues such as subcircuit name mapping. These restrictions have been lifted with the direct interface.

There are several advantages of the direct simulator integration approach over the socket simulator integration approach, namely:

- **Improved Performance in Netlisting**

  Netlisting is much faster in the direct approach, the direct approach supports incremental netlisting. This ensures enhanced performance when incremental updates are performed in a design and then netlisted.

- **Better Readability of Netlists**

  The netlists are truly hierarchical and all numeric values in the netlist are more readable. For example in *Hspice* socket, the numeric values are changed from -5.0 to -5.0000000 in the final netlist. Also, the sub-circuits are no longer unfolded. The sub-circuits are also no longer mapped unless necessary.

  To customize the format of a netlist file, you can set the following SKILL variables in your `.cdsinit` file or CIW:

  - `hspiceSoftLineLength` variable to control the maximum length of a line of netlist output after which line is automatically split into multiple lines for convenient reading. You can set this variable to an integer value less than or equal to 1024 characters. This variable overrides the OSS variable, `softLineLength`, which is set to 1024 by default.
  - `hspiceMaxLineLength` variable to increase or decrease the maximum limit on the number of characters to be printed in a line of netlist output from the default. This variable overrides the OSS variable, `maxLineLength`. The `hspiceSoftLineLength` value can never be greater than `hspiceMaxLineLength` value.

  For more information, see the *Customizing the Hierarchical Netlister (HNL)* chapter of the *Open Simulation System Reference*.

  You can use HSPICE reserved words, temper and hertz, as design variable names in the netlist without OSS remapping them. For more information, see the *hnlReservedNameList* section of the *Open Simulation System Reference*.

- **Read-only Designs can be Simulated, Provided they are Extracted**

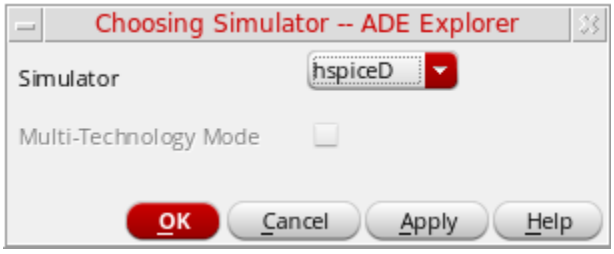

© 2023 Cadence Design Systems, Inc.

A limitation of socket netlisting is that the top cell of a design needs to be editable before the design can be netlisted. The direct approach however, allows read only designs to be simulated.The only pre-requisite being, that the design needs to be extracted first, so that connectivity information is written to the database.

- **Advanced Evaluation of Operators**

  Direct netlisting supports the evaluation of ternary operators (Example, `(iPar("r")>2e-3?200e-3:400e-3`), whereas the same is not supported by socket netlisting.

To set hspiceD, choose *Setup – Simulator*. The *Choosing Simulator -- ADE Explorer* form appears. Select `hspiceD` simulator in this form.




The *Virtuoso Analog Design Environment* window displays with the *hspiceD* simulator selected:

## Libraries

The following cells of the `analogLib` library are updated to contain *HspiceD* views. The *HspiceD* simInfo, CDF parameters and netlisting procedures have been added to all these `analogLib` cells. For more information, see *Analog Librray Reference*.

| | | | | | |
|---|---|---|---|---|---|
| bcs | bvs | cap | cccs | ccvs | core |
| diode | iam | idc | iexp | ind | iopamp |
| ipulse | ipwl | ipwlf | npn | isffm | isin |
| ixfmr | nbsim | nbsim4 | njfet | nmes | nmes4 |
| nmos | nmos4 | pbsim | pbsim4 | pcapacitor | pdiode |
| pjfet | pmos | pmos4 | pnp | presistor | res |
| schottky | vam | tline | u1wire | u2wire | u3wire |
| u4wire | u5wire | usernpn | userpnp | vccap | vccs |


© 2023 Cadence Design Systems, Inc.

| | | | | | |
|---|---|---|---|---|---|
| vcres | vcvs | vdc | vexp | vpulse | vpwl |
| vpwlf | vsffm | vsin | winding | xfmr | zener |
| iprobe | pinductor | mind | pmind | pvccs2 | pvccs3 |
| pvcvs | pvcvs2 | pvcvs3 | pvccs | | |


© 2023 Cadence Design Systems, Inc.

# HspiceD Features

The use model of the Analog Design Environment for the *HspiceDirect* simulator is similar to that of the *Spectre Direct*/*Hspice Socket* interface. Most of the options work in the same way with a few differences.

## Model Libraries

Similar *Model Library Setup* form is applied for HSPICE simulator. You can enter model file names into the *Model Library File* field. The list box displays the list of model files to be included. You can also include an optional *Section* field. When the *Section* field for a particular model file is defined, the netlist contains the following statement:

```
.LIB "<modelLibraryFile>" <section>
```

When the *Section* field is not defined, the netlist will contain the statement,

```
.INCLUDE "<modelLibraryFile>"
```

For detailed information about the form, refer to the section Choosing Model Library Path on page 99.

## Distributed Processing Support

The Distributed Processing mode is supported only for normal simulation and parametric analysis. For detailed information about Distributed Processing, refer to the *Virtuoso Analog Distributed Processing Option User Guide*.

## Running Analyses

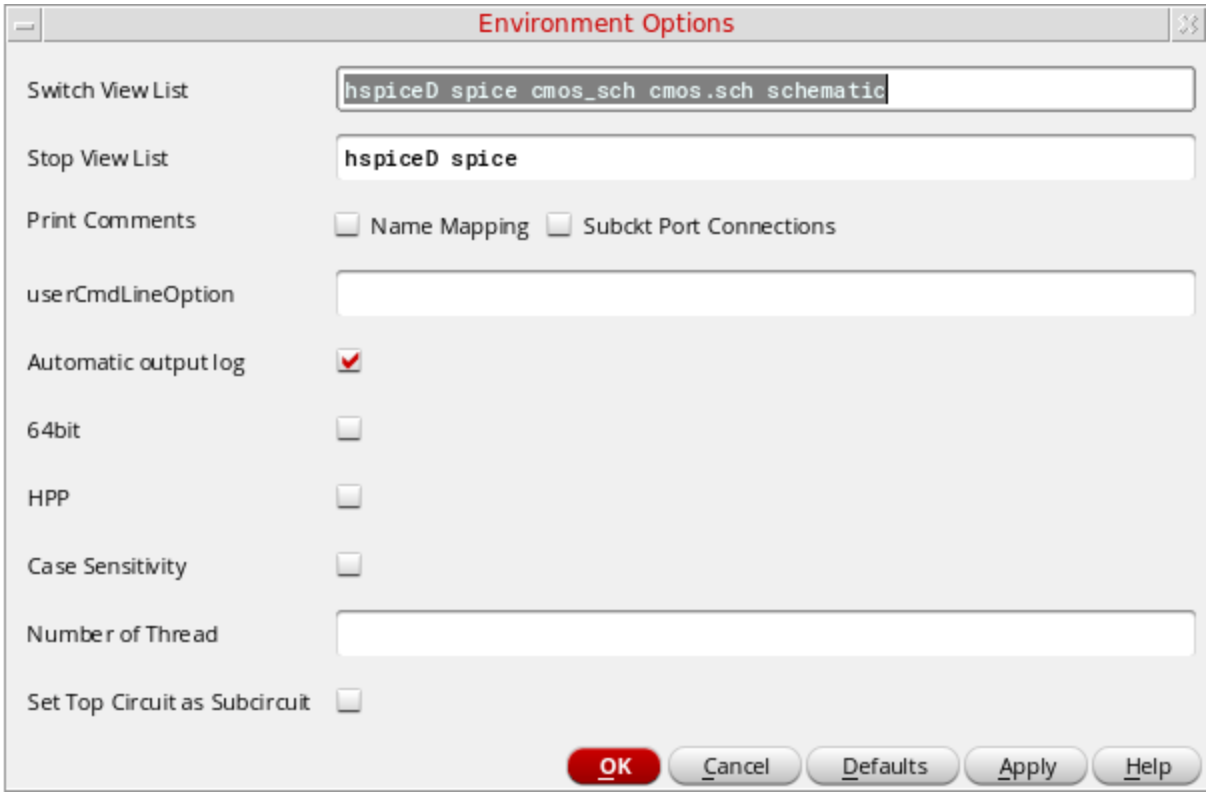
The *Choosing Analyses* form helps you to set up and run an analysis. This form is explained in details in the Setting up Analysis chapter of this book. Refer to this section for details about each analysis.

The analyses that are supported are: *DC, Transient, AC, Noise* and *OP*. To run an analysis, select it in the *Choosing Analyses* form. The form re-displays to show the fields that are required for the selected analysis.


© 2023 Cadence Design Systems, Inc.

## Passing Command-Line Options

You can pass the command-line options using the *userCmdLineOption* option in the *Environment Options* form.




Alternatively, you can pass the command-line options using the `userCmdLineOption` environment variable in the `.cdsinit` file as follows:

```
envSetVal("hspiceD.envOpts" "userCmdLineOption" 'string "string value")
```

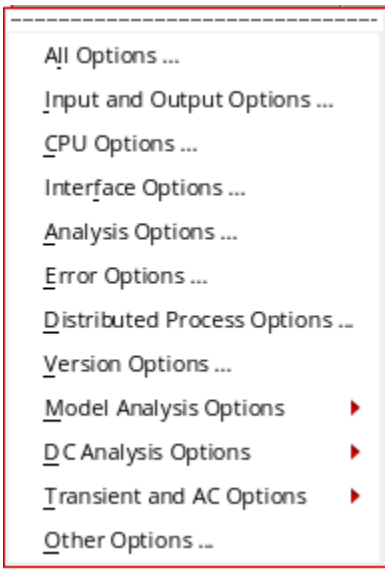
For example, to set multithreading to `3`, you should set the `userCmdLineOption` environment variable as shown below:

```
envSetVal("hspiceD.envOpts" "userCmdLineOption" 'string "-mt 3")
```


© 2023 Cadence Design Systems, Inc.

## Specifying HSPICE Options

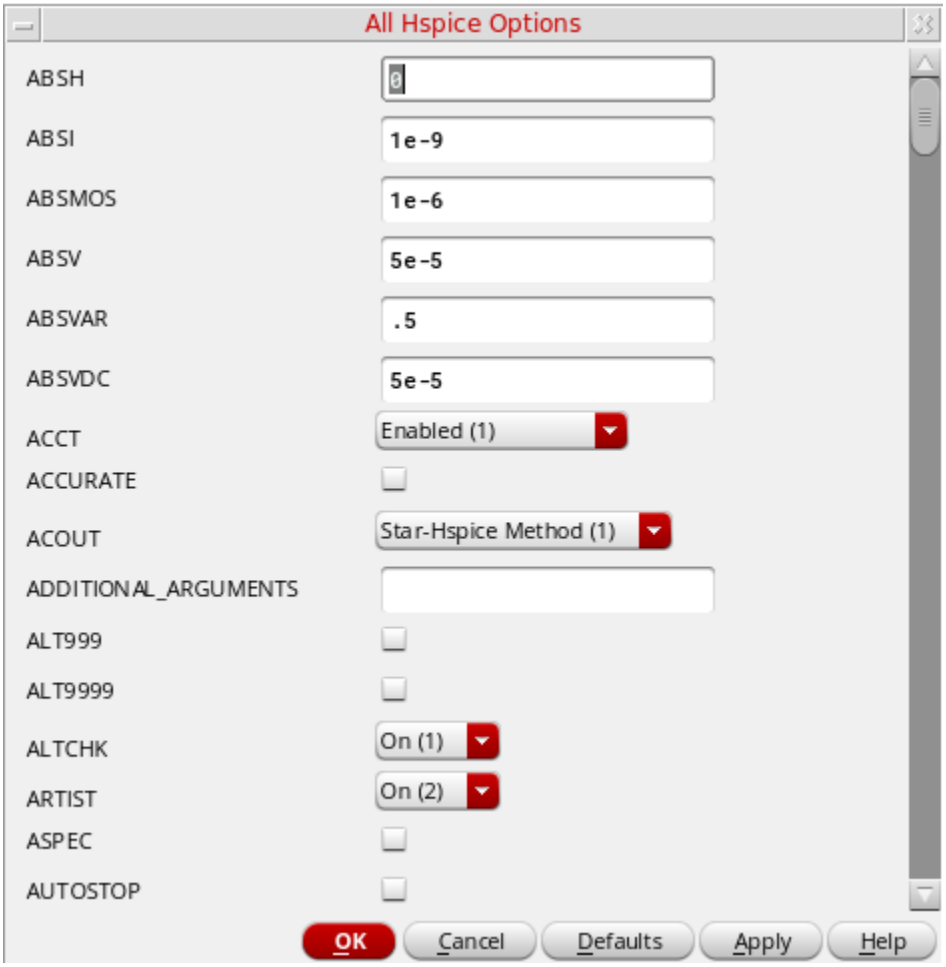
You can specify appropriate HSPICE simulator options using *Simulation – Analog Options.* These options can be used to modify various aspects of the simulation, including output types, accuracy, speed, and convergence.





© 2023 Cadence Design Systems, Inc.

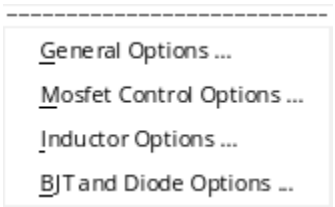
The below form appears when you click *All Options*.




For details about the Analog Options, refer to *HSPICE/SPICE Interface Reference*.

### Model Analysis Options

The *Model Analysis Options* have been grouped as follows:



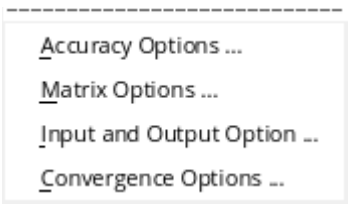


© 2023 Cadence Design Systems, Inc.

- Click *Model Analysis Options – General Options* to specify DCAP, HIER_SCALE, MODMONTE, MODSRH, SCALE, TNOM using the *Hspice General Model Options* form.
- Click *Model Analysis Options – Mosfet Control Options* to specify CVTOL, DEFAD, DEFAS, DEFEL, DEFNRD, DEFNRS, DEFPD, DEFPS, DEFW, SCALM, WL using the *Hspice Mosfet Control Options* form.
- Click *Model Analysis Options – Inductor Options* to specify GENK, KLIM using the *Hspice Inductor Options* form.
- *Model Analysis Options – BJT and Diode Options* to specify EXPLI using the *Hspice BJT and Diode Options* form.

### DC Analysis Options

The *DC Analysis Options* have been grouped as follows:

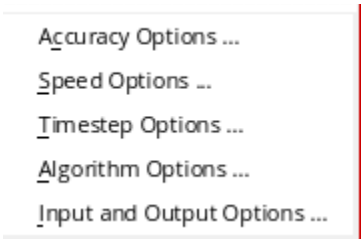



- Click *DC Analysis Options – Accuracy Options* to specify ABSH, ABSI, ABSMOS, ABSVDC, DI, KCLTEST, MAXAMP, RELH, RELI, RELMOS, RELV, RELVDC using the *Hspice DC Accuracy Options* form.
- Click *DC Analysis Options – Matrix Options* to specify ITL1. ITL2, NOPIV, PIVOT, PIVREF, PIVREL, PIVTOL using the *Hspice Matrix Options* form.
- Click *DC Analysis Options – Input and Output Option* to specify CAPTAB, DCCAP, VFLOOR using the *HspiceDC Input and Output Options* form.
- Click *DC Analysis Options – Convergence Options* to specify CONVERGE, CSHDC, DCFOR, DCHOLD, DCON, DCSTEP, DV, GMAX, GMINDC, GRAMP, GSHUNT, ICSWEEP, ITLPTRAN, NEWTOL, OFF, RESMIN using the *HspiceConvergence Options* form.


© 2023 Cadence Design Systems, Inc.

### Transient and AC Options

The *Transient and AC Analysis Options* have been grouped as follows:




- Click *Transient and AC Options – Accuracy Options* to specify ABSH, ABSV, ACCURATE, ACOUT, CHGTOL, CSHUNT, DI, GMIN, GSHUNT, MAXAMP, RELH, RELI, RELQ, RELV, RISETIME, TRTOL using the *Hspice Transient and AC Options* form.
- Click *Transient and AC Options – Speed Options* to specify AUTPSTOP, BKPSIZ, BYPASS, BYTOL, FAST, ITLPZ, MBYPASS, TRCON using the *Hspice Speed Options* form.
- Click *Transient and AC Options – Timestep Options* to specify ABSVAR, DVDT, FS, FT, IMAX, IMIN, ITL5, RELVAR, RMAX, RMIN, SLOPETOL, TIMERES using the *Hspice Timestep Options* form.
- Click *Transient and AC Options – Algorithm Options* to specify DVTR, IMAX, IMIN, LVLTIM, MAXORD, METHOD, MU, PURETP, TRCON using the *Hspice Algorithm Options* form.
- Click *Transient and AC Options – Input and Output Options* to specify INTERP, ITRPRT, MEASFAIL, MEASSORT, PUTMEAS, UNWRAP using the *Hspice Transient Input and Output Option*s form.

*Important*

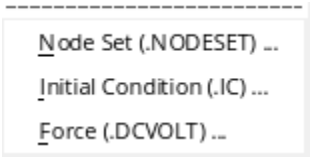
All the *Analog Options* mentioned are supported by the *Hspice* version `2003.3`. Ensure that you are using a compatible version of *Hspice.*

## Output Log

This displays the file `hspice.out` found under the `psf` directory. This is the file to which the HSPICE output is re-directed.


© 2023 Cadence Design Systems, Inc.

## Convergence Aids



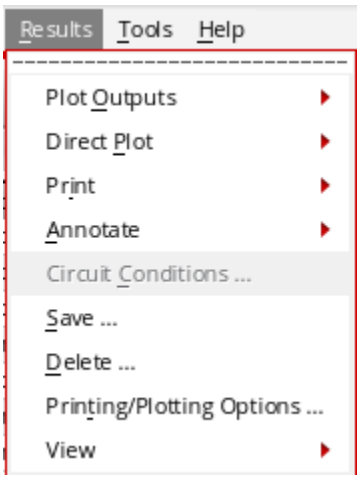

Click *Convergence Aids – Node Set (.NODESET)* to initialize specified nodal voltages for a DC operating point analysis. The `.NODESET` statement is used to correct convergence problems in a DC analysis. Setting nodes in the circuit to values that are close to the DC operating point solution enhances the convergence of the simulation. The *Select Node Set* form works in the same way as the Spectre Direct/HSPICE Socket interface. The netlist will contain the `.NODESET` statement line.

Click *Convergence Aids – Initial Condition (.IC*) or *Convergence Aids – Force (.DCVOLT)* to set the transient initial conditions. The initialization depends on whether the UIC parameter is included in the .TRAN analysis statement. If the UIC parameter is specified in the .TRAN statement, the HSPICE simulator does not calculate the initial DC operating point. Consequently, the transient analysis is entered directly.

The *Select Initial Condition Set* and the *Select Force Node Set* forms work in the same way as the *Spectre Direct/Hspice Socket* interface. The netlist contains the `.IC` and the `.DCVOLT` statement line, whichever the case may be.

## Results

You can save, select, delete, restore, plot and print a set of simulation results using the *Results* menu.





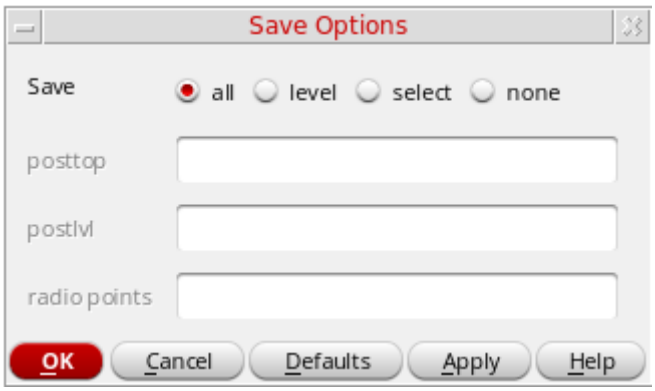
© 2023 Cadence Design Systems, Inc.

The following menus have been removed from the *Hspice Direct* interface as the *Hspice* simulator does not write the specified data in the `psf` files:

- *Plot Outputs – Noise*
- *Direct Plot – Equivalent Output Noise*
- *Direct Plot – Equivalent Input Noise*
- *Direct Plot – Squared Output Noise*
- *Direct Plot – Squared Input Noise*
- *Direct Plot – Noise Figure*
- *Print – Model Parameters*
- *Print – Noise Parameters*
- *Print – Noise Summary*
- *Annotate – Model Parameters*

The noise data is written by the *HSPICE* simulator in the `hspice.out` file. Use the menu *Simulation – Output Log* to view the simulator output file.

To specify the save options for the outputs, choose *Outputs – Save All*. The *Save Options* form appears.




In this form, select one of the following options to specify the output data you want to save:

- *All*—Select this option to save all the outputs. After you select this option, the *posttop*, *postlvl*, and *radio points* fields become unavailable.
- *Level*—Select this option to specify the levels upto which you want the simulation results data to be saved. After you select this option, the *posttop* and *postlvl* options become available.


© 2023 Cadence Design Systems, Inc.

- *posttop*—Specify the value up to which you want the data to be saved. For example, if you specify the value as `3`, the data for only the top three level nodes is saved in the results file.
- *postlvl*—Specify the value for which you want the data to be saved. For example, if you specify the value as `2`, the data for only the second-level node (voltage and current) is saved.

- *select*—Select this option to specify the radio points that you want to save. After you select this option, the *radio points* option becomes available, where you can individually specify the selective voltage or current nets that you want to save, separated by a space character, as shown below:

  ```
  v(v1) v(v2) i(i1) i(i2)
  ```

- When you generate the netlist, these nets appear in `.probe` statement.
- *none*—Select this option if you do not want to save any data.

## Control Mapping of Nets

In HspiceD netlister, you can control mapping of `gnd!` nets by setting an environment variable as shown below:

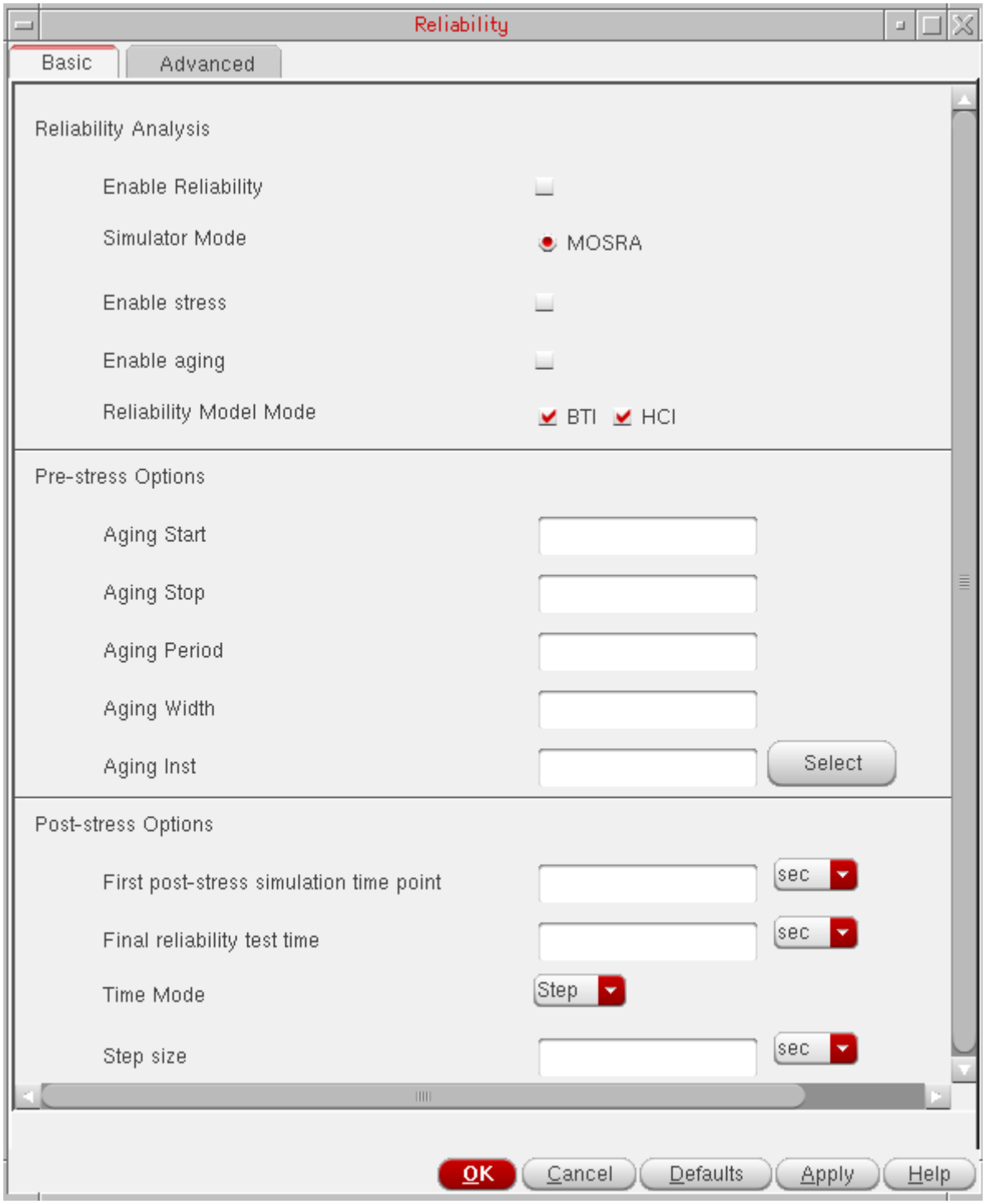
```
envSetVal("hspiceD.envOpts" "mapGndNetToZero" 'boolean t)
```

The default value of this environment variable is `t` implying that by default, the hspiceD netlister maps `gnd!` nets to 0. On setting this variable to `nil`, the HspiceD netlister stops mapping `gnd!` nets to 0. You can set the variable to `nil` in .cdsinit file.

## Setting up Reliability Option

To set up the reliability analysis options, choose *Simulation – Reliability – Setup*. The form appears. When you select HSPICE as simulator, the form displays two tabs – *Basic* and *Advanced*.

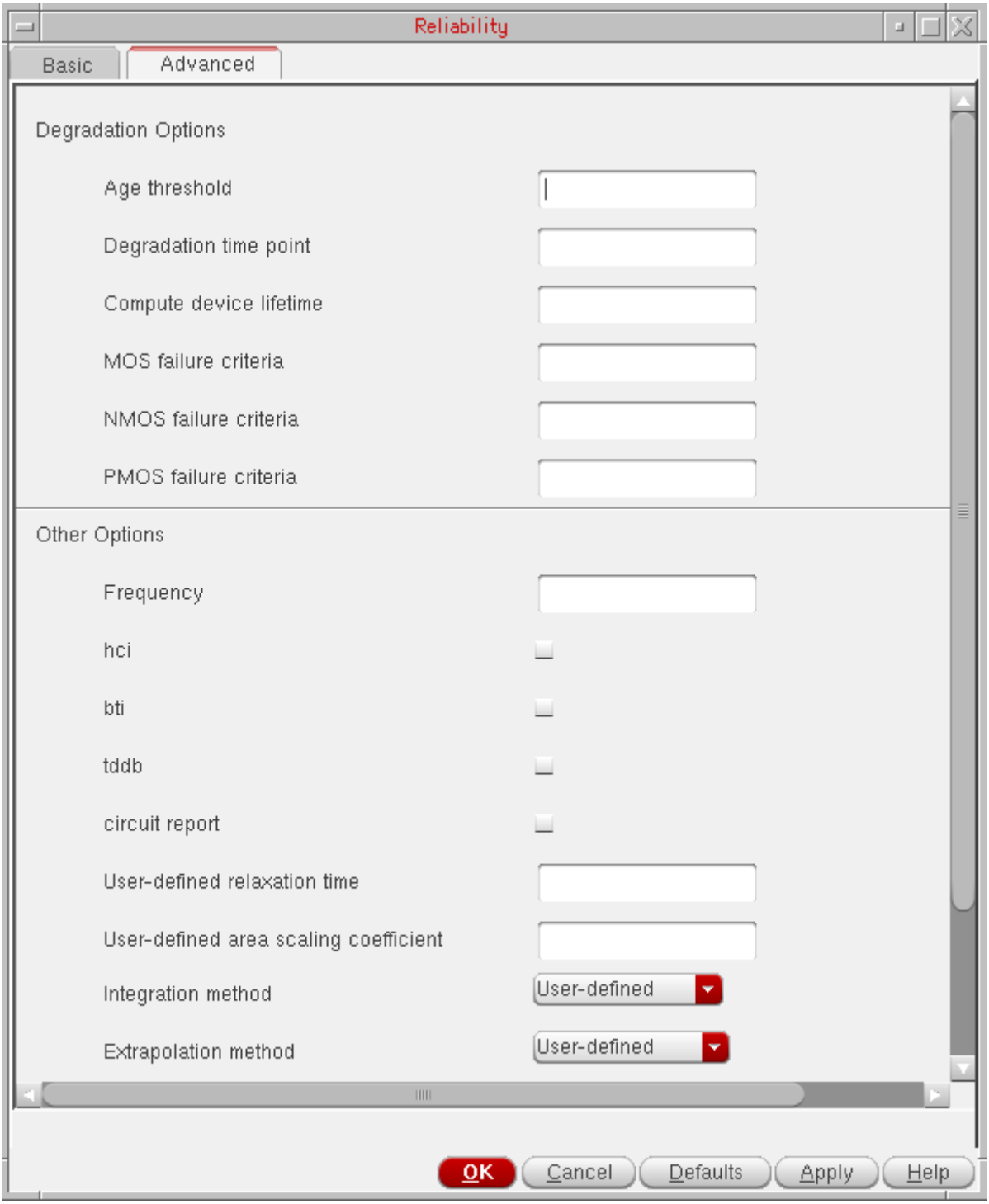

© 2023 Cadence Design Systems, Inc.





© 2023 Cadence Design Systems, Inc.

The *Advanced* tab shows the following fields:





© 2023 Cadence Design Systems, Inc.

© 2023 Cadence Design Systems, Inc.

# 17

# Simulating Multi-Technology Designs and Defining Locally Scoped Models and Options

You can use multi-technology simulation (MTS) when designing high-speed interfaces between integrated circuits (ICs) to see the effects of signals traveling between two or more chips. You can also use MTS for custom IC system-in-package (SiP) designs. SiP designs are fully functional systems or subsystems in an IC package. MTS helps you combine passive components with ICs fabricated using different process technologies while minimizing parasitic loading.

The MTS design should consist of a top-level schematic that contains two or more blocks, each with its own associated set of process definitions, model libraries, simulation states, and other setup files. You cannot use MTS with a non-hierarchical schematic because each different technology must be self-contained. You must instantiate device instances from different technologies in different cellviews so that the program can map model information appropriately.

See also:

- Enabling Locally Scoped Models and Options
- Enabling Local Scoping for a Cell
  - Specifying Locally Scoped Models for a Cell
  - Specifying Local Simulator Options for a Cell
- Enabling Local Scoping for an Instance
  - Specifying Locally Scoped Models for an Instance
  - Specifying Local Simulator Options for an Instance

*Important Points to Note*

- Different MTS setups can be stored for different tests.


© 2023 Cadence Design Systems, Inc.

- To set up MTS options, the design in your test must be a config view.

Video

You can watch a video demonstration for this feature at Multi-Technology Simulation (MTS) with Local Scoped Models. Also, read the related blog at Virtuoso Video Diary: Multi-Technology Simulation–The Good has Changed for Better.

# Enabling Locally Scoped Models and Options

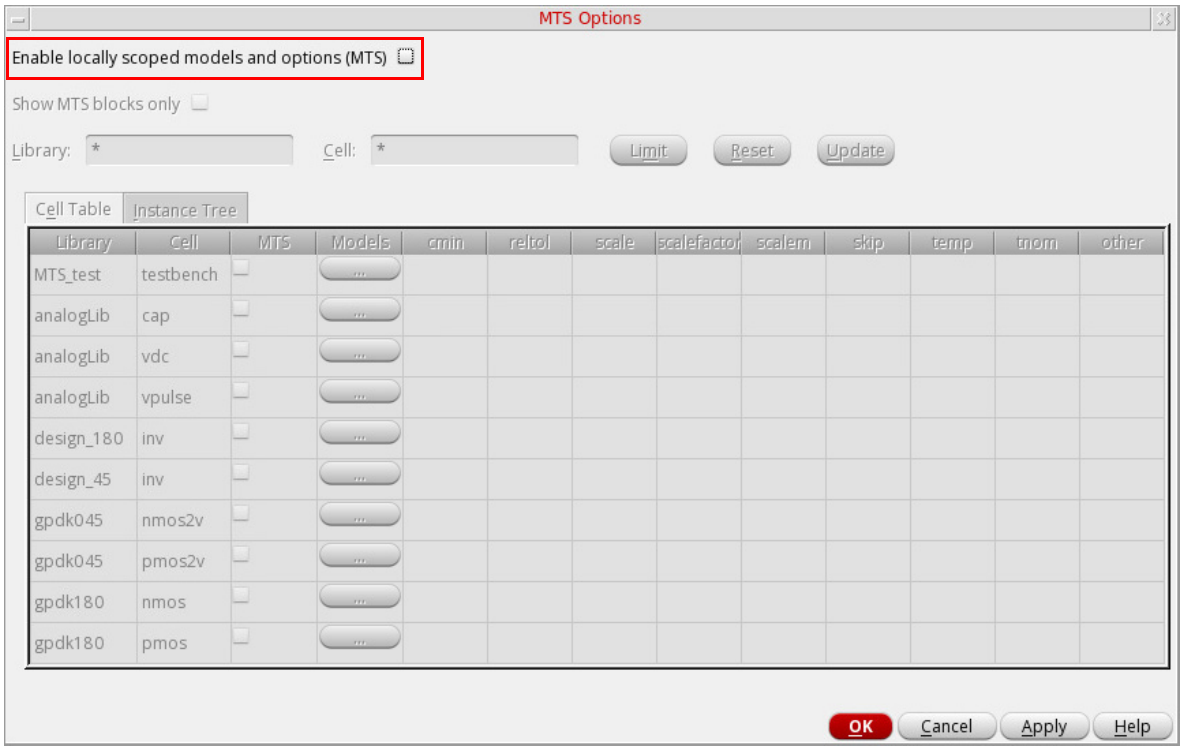
*Important*

You cannot enable locally scoped models and options (MTS options) for third-party simulators.

To enable MTS options in your test setup, do the following:

1. In the *Setup* assistant, right-click the test name, and then choose *MTS Options*.

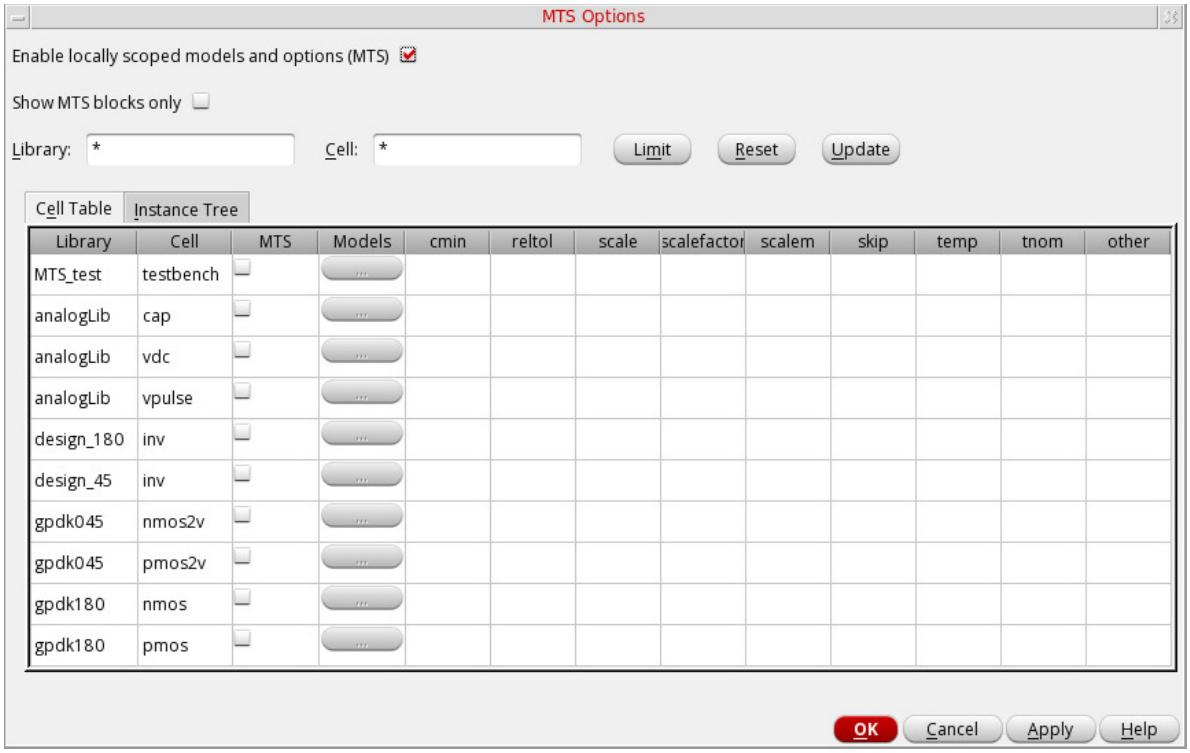
   The MTS Options form appears.





© 2023 Cadence Design Systems, Inc.

2. Select the *Enable Locally scoped models and options (MTS)* check box to enable the MTS options for the selected test.




3. Click *OK* to save the changes.

After enabling MTS options in your test setup, you can enable local scoping at the following levels in your design hierarchy:

- Cell
- Instance

### *Important Points to Note*

- For Spectre, UltraSim, and AMS simulators, local scoping can be enabled at both these levels– the cell and instance levels.
- For third-party simulators, you cannot enable MTS options in your test setup.

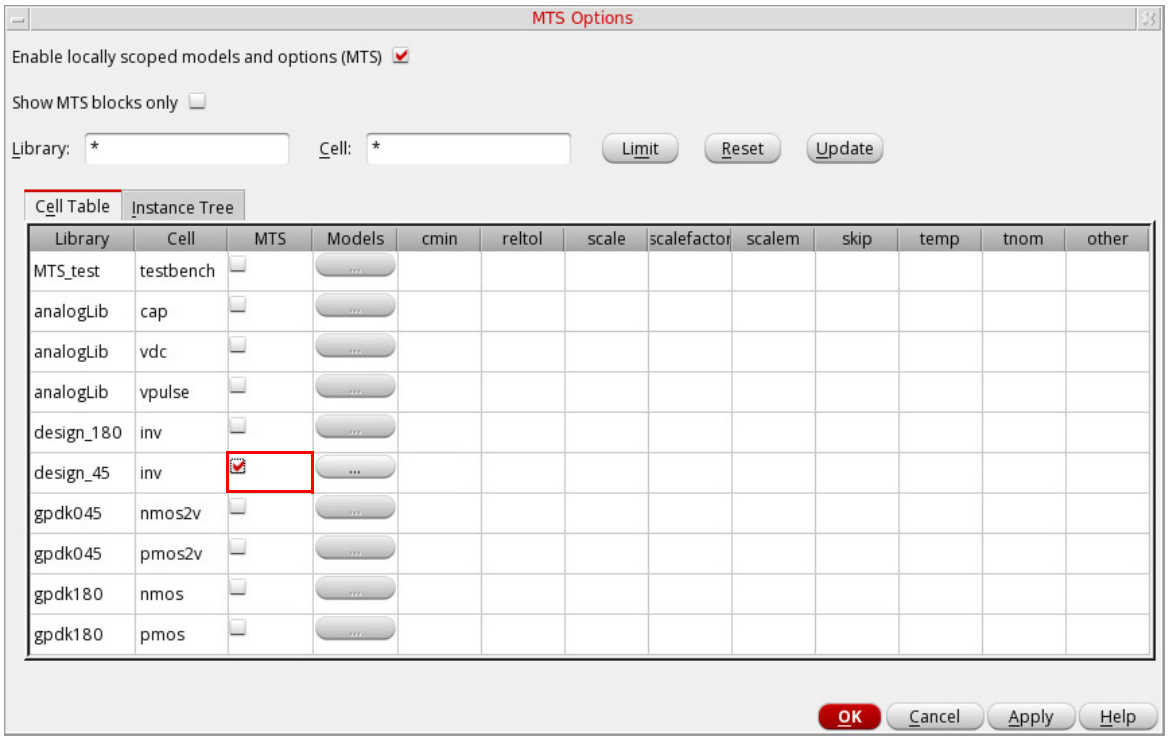

© 2023 Cadence Design Systems, Inc.

## Enabling Local Scoping for a Cell

**Note:** Before enabling local scoping at the cell level, ensure that you have enabled local scoping in your test setup. See the Enabling Locally Scoped Models and Options section for more information.

To enable local scoping for a cell, do the following:

1. In the MTS Options form, click the *Cell Table* tab to view a list of the library and cells in your test setup.
2. Click the check box in the *MTS* column for the cell that you want to enable for local scoping, as shown in the following figure.




3. Click *OK* to save the changes.

   The selected cell is now enabled for local scoping.

   You can now specify the model libraries that will be locally scoped at the selected cell. See the Specifying Locally Scoped Models for a Cell section for more information.

   You can also specify various process parameters that you want to include locally in the simulation. See the Specifying Local Simulator Options for a Cell section for more information.


© 2023 Cadence Design Systems, Inc.

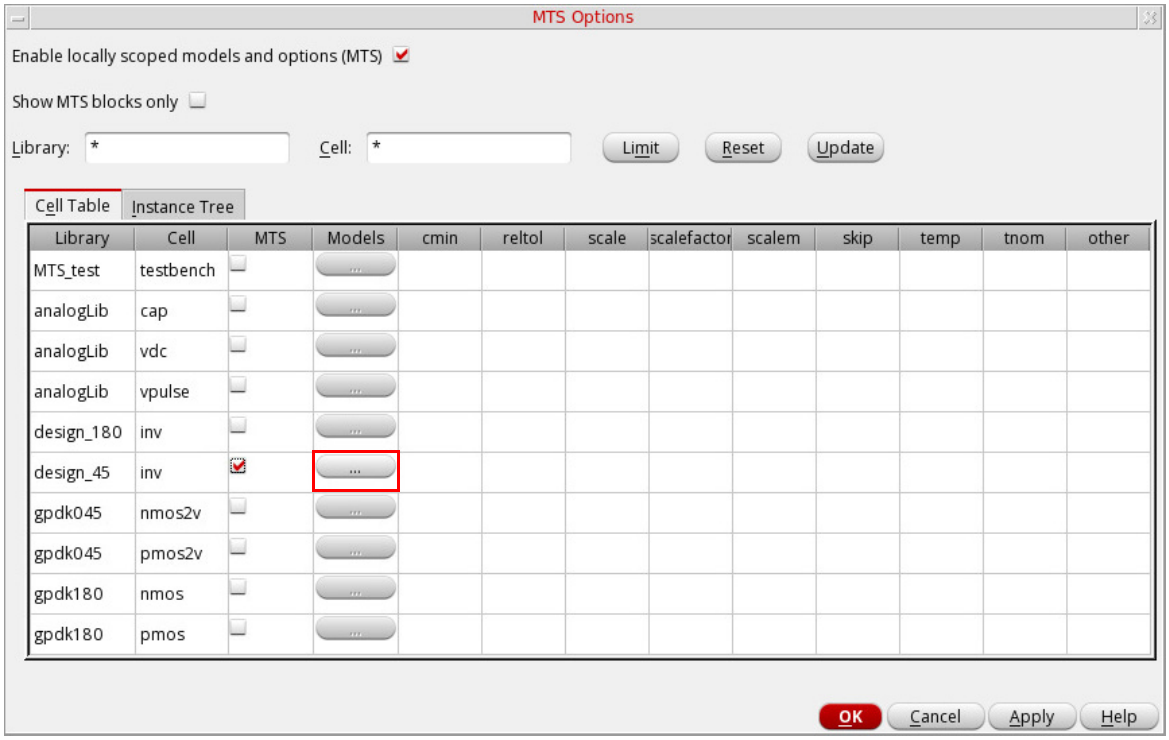
## Specifying Locally Scoped Models for a Cell

*Important*

You can specify locally scoped models only for the cells that are enabled for local scoping. See the Enabling Local Scoping for a Cell section for more information.

To specify the models that will be locally scoped at the cell level, do the following:

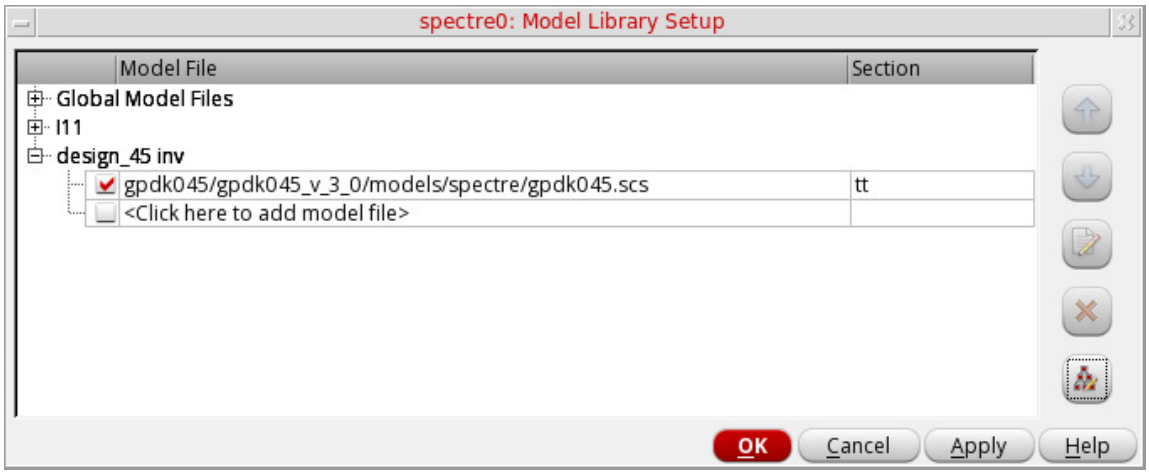
1. On the *Cell Table* tab, in the row for the cell for which you have enabled local scoping, click the ellipsis (…) button the *Models* column, as shown in the following figure.





© 2023 Cadence Design Systems, Inc.

The Model Library Setup form appears, as shown in the following figure.




2. Follow the instructions described in the Choosing Model Library Path section to select the model files.


© 2023 Cadence Design Systems, Inc.

## Specifying Local Simulator Options for a Cell

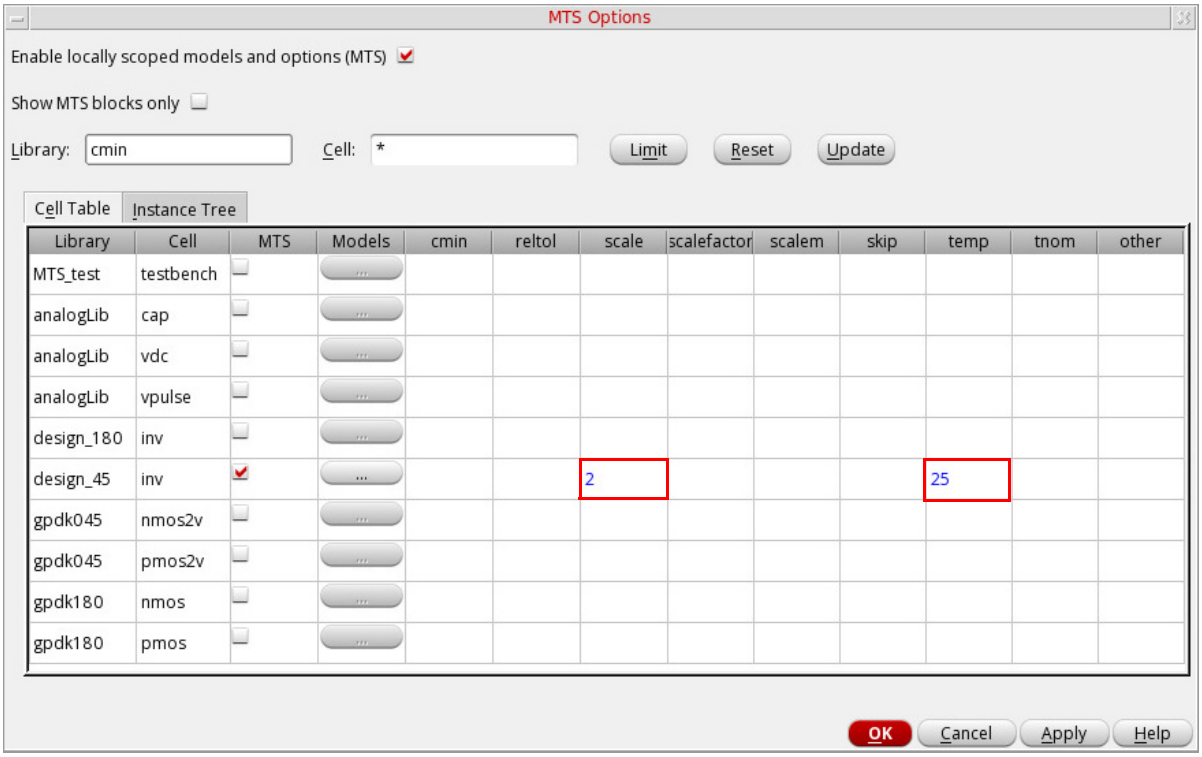
*Important*

You can specify local simulator options only for the cells that are enabled for local scoping. See the Enabling Local Scoping for a Cell section for more information.

To specify the various process parameters, such as `tnom`, `scale`, `scalefactor`, `temp`, and `scaleem`, that you want to include in the simulation for a cell, do the following:

1. On the *Cell Table* tab, click in the column for the simulator option you want to specify, and then type a value.

   The following figure shows an example where the local simulator options, `scale` and `temp`, are specified for the `inv` cell.




   **Note:** You can also specify a `VAR` expression in the *scale*, *scalem*, *scalefactor*, *temp*, and *tnom* columns. The variable name specified in the `VAR` expression is added to the *Design Variables* section of *Setup* assistant.

2. Click *OK* to save the changes.


© 2023 Cadence Design Systems, Inc.

***Important points to note***

- Currently, all the supported Spectre options are provided as columns, such as, `cmin`, `reltol`, `tnom`, `scale`, `scalefactor`, `scalem`, `skip`, `temp`, and `tnom`, in the MTS Options form. If you want to include these options in the simulation locally, you can specify them in the respective columns.
- In addition to the currently supported local options, if Spectre supports new local options in future, you can include them too in the simulation locally by specifying their values in the `other` column.

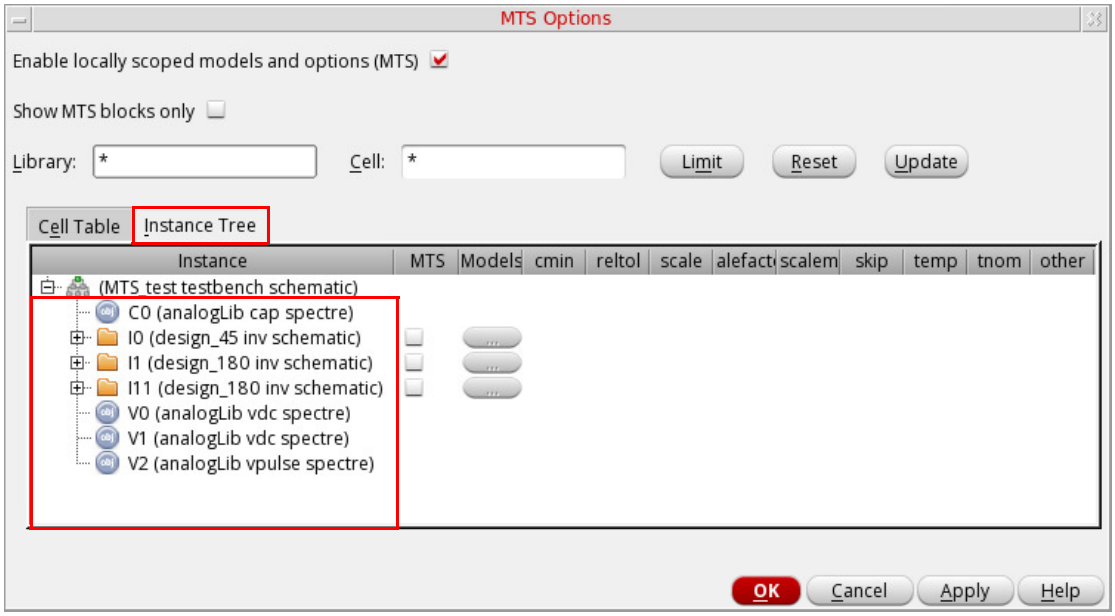
## Enabling Local Scoping for an Instance

*Important*

Before enabling local scoping at the instance level, ensure that you have enabled it in your test setup. See the Enabling Locally Scoped Models and Options section for more information.

To enable local scoping for an instance, do the following:

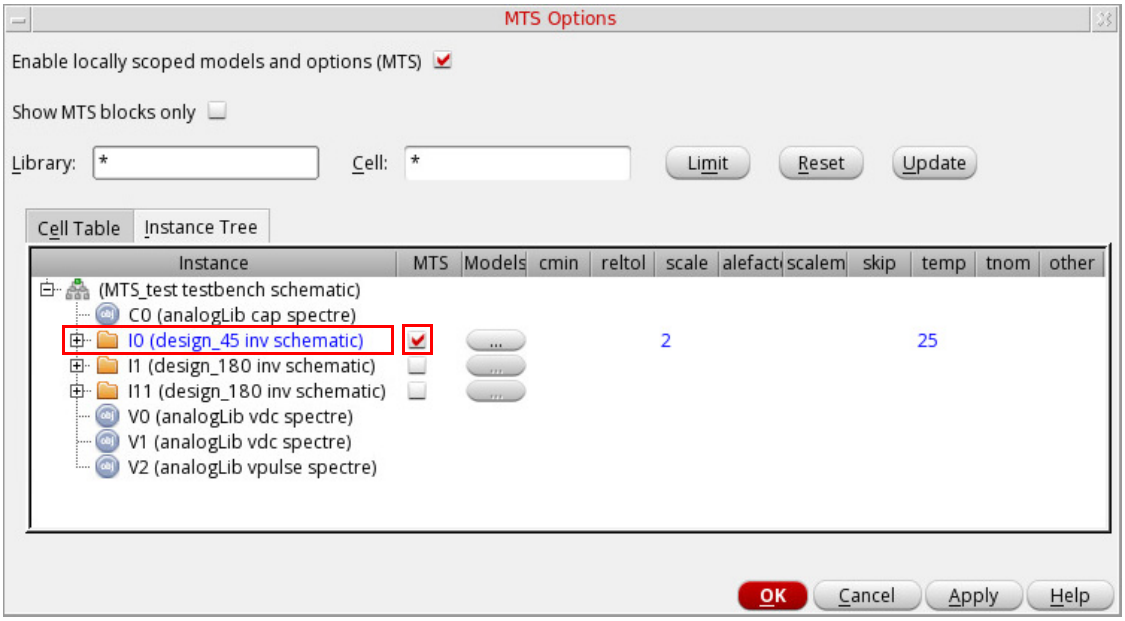
1. In the *MTS Options* form, click the *Instance Tree* tab to view an alphabetical list of the instances available in your test.





© 2023 Cadence Design Systems, Inc.

2. Select the check box in the *MTS* column for the instance that you want to enable for local scoping.




**Note:** Instances that appear in blue indicate that local scoping has been enabled for them.

3. Click *OK* to save the changes.

You can now specify the model libraries that will be locally scoped for the selected instance. See the Specifying Locally Scoped Models for an Instance section for more information.

You can also specify various process parameters that you want to include locally in the simulation. See the Specifying Local Simulator Options for an Instance section for more information.


© 2023 Cadence Design Systems, Inc.

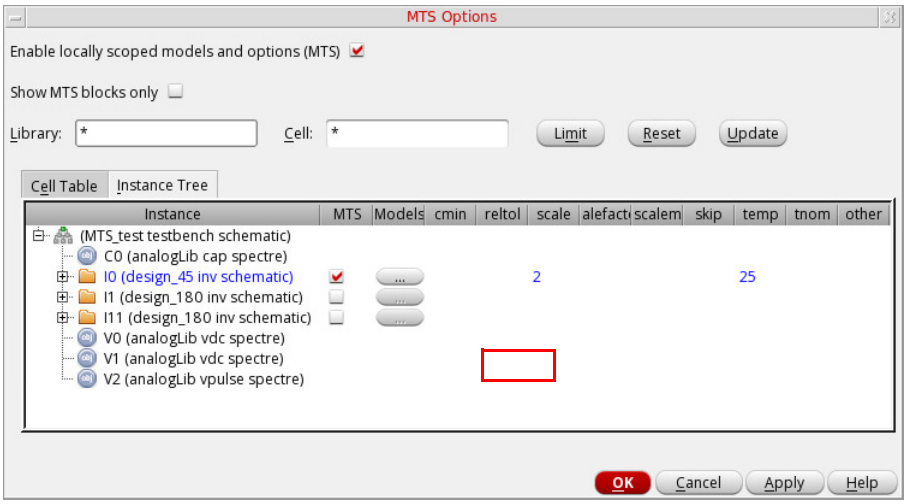
## Specifying Locally Scoped Models for an Instance

*Important*

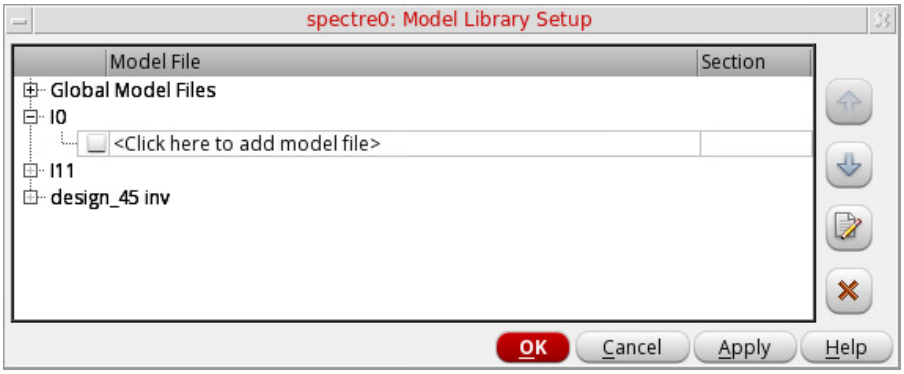
You can specify locally scoped models only for the instances that are enabled for local scoping. See the Enabling Local Scoping for an Instance section for more information.

To specify the models that will be locally scoped for an instance, do the following:

1. On the *Instance Tree* tab, click the ellipsis (...) button in the *Models* column for the instance that is enabled for local scoping. Such instances appear in blue.




The Model Library Setup form appears, as shown in the following figure.




2. Follow the instructions described in the "Specifying Model Libraries" section to select the model files.


© 2023 Cadence Design Systems, Inc.

## Specifying Local Simulator Options for an Instance

*Important*

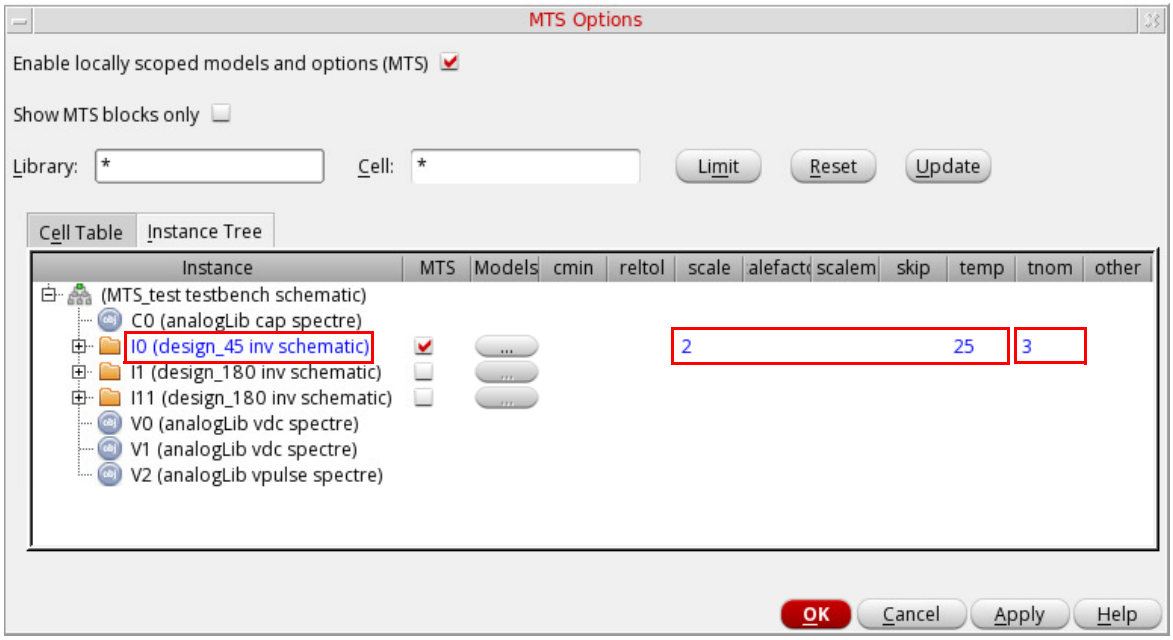
You can specify local simulator options only for the instances that are enabled for local scoping. See the Enabling Local Scoping for an Instance section for more information.

To specify the various process parameters, such as `tnom`, `scale`, `scalefactor`, `temp`, and `scaleem`, that you want to include in the simulation for an instance locally, do the following:

1. On the *Instance Tree* tab, click in the column for the simulator option you want to specify, and then type a value.

   **Note:** If the local simulator options are specified at the cell level, they are copied to the instances from their cells, provided that the local scoping is enabled at both the cell and the instance levels.

   The following figure shows an example where the local simulator option, `tnom`, is specified at the `I0` instance and `scale` and `temp` are copied from the `inv` cell that contains the `I0` instance in the design hierarchy.




2. Click *OK* to save the changes.


© 2023 Cadence Design Systems, Inc.

## Disabling Local Scoping

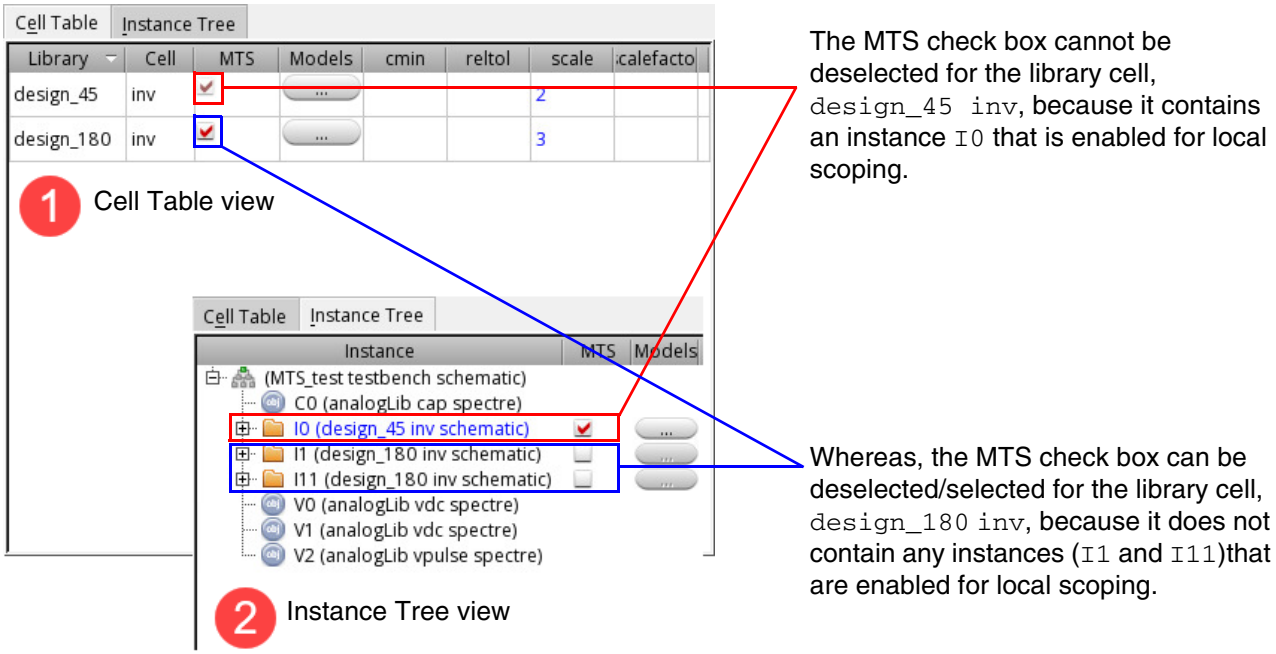
Perform the following steps to disable the local scoping for a cell or an instance:

- In the MTS Options form, on the *Cell Table* tab or the *Instance Tree* tab, deselect the *MTS* check box next to a cell or an instance for which you want to disable the local scoping.

### *Important Point to Note*

You cannot disable the local scoping for a cell if the local scoping is enabled for any of its instances. For such library cells the *MTS* check box appears dimmed and cannot be deselected.

In the following example, a side-by-side view of the *Cell Table* and *Instance Tree* views is shown.





© 2023 Cadence Design Systems, Inc.

## Limiting the View by Library or Instance

You can limit what appears on the MTS Options form by library name, cell name, or both.

**Note:** You can reset the filter to view all cells in all libraries by clicking *Reset*.

To limit the view of design components on the MTS Options form to only those in a particular library or set of libraries, do the following:

1. In the *Library* field, type a filtering string.
2. Click *Limit*.

   Only those design libraries that match the filtering string are displayed in the *MTS Options* form.

To limit the view of design components on the *MTS Options* form to a particular cell or set of cells, do the following:

1. In the *Cell* field, type a filtering string.
2. Click *Limit*.

   Only those cells that match the filtering string are displayed in the MTS Options form.

To limit the view by library name and by cell name, do the following:

1. In the *Library* field, type a filtering string.
2. In the *Cell* field, type a filtering string.
3. Click *Limit*.

   Only those cells that match the filtering string in the *Cell* field contained in those libraries that match the filtering string in the *Library* field are displayed in the MTS Options form.
4. Select the *Show MTS blocks only* check box to display only MTS Blocks in the *Cell Table*. The MTS blocks are indicated with a check mark in the *MTS* column of the *Cell Table*
5. Click the *Reset* button to clear the filtering and show all libraries and cellviews.

Click the *Update* button view only the blocks that are selected as MTS blocks.


© 2023 Cadence Design Systems, Inc.

© 2023 Cadence Design Systems, Inc.

# 18

# auLvs Netlisting

This appendix briefly describes Analog LVS or auLvs (Analog and Microwave Layout Versus Schematic) netlister. It is the analog and microwave version of LVS, which originally ran only on digital designs. This information is applicable to any 4.4 or above versions of the Virtuoso® design framework II (DFII).

## Using auLvs

You use the auLvs tool for designs that depend on CDF and AEL information and when you use the Analog Design Environment. You can run auLvs inside or outside the DFII environment.

To translate files from the DFII database format into an auLvs netlist, follow the steps below:

1. Set the CDS_Netlisting_Mode variable in the `.cshrc` file to `Analog` or `Compatibility` so that the Analog LVS tool (auLvs) is used.The syntax for this variable is

   ```
   setenv CDS_Netlisting_Mode "{Analog|Compatibility}"
   ```

2. Create an auLvs view for the cell by copying the symbol view to the auLvs view.
3. Add the auLvs simulation information to the cell's CDF.

## Customization Using the .simrc File

You can further control the behavior of the netlist by using the simulation run control (`.simrc`) file. The parameters that you can include in the `.simrc` file are the same as those that are defined using the `simSetDef SKILL` function. This SKILL function defines variables only if they have not been defined previously (that is, during initialization when the `si.env` and `.simrc` files are read).


© 2023 Cadence Design Systems, Inc.

The following auLvs parameter can be set in the `.simrc` file:

| Parameter | Description |
|---|---|
| `lvsLimitLinesInOutFile` | You can set this to an integer value, the default being 20. If the file `si.out` contains more than the number specified, the path of the file with the informative message is written into the `si.out` file; otherwise, the contents are written into the *si.out* file. |

# Related Documentation on auLvs

| For information on | See the following Cadence documents |
|---|---|
| Adding CDF information for auLvs | The topic *Adding Component Description Format Simulation Information* in the Virtuoso Parasitic Aware Design User Guide |
| Where the auLvs view (the default stopping view for auLvs) is required in a parasitic simulation | The topic *Adding Component Description Format Simulation Information* in the Virtuoso Parasitic Aware Design User Guide |
| The auLvs SKILL function | The chapter *Netlist Functions* in Virtuoso Analog Design Environment L SKILL Reference |
| ■ Simulator options applicable to auLvs | The topic *Modifying Simulation Information* in the Component Description Format User Guide |
| ■ SKILL expression to set the options for the auLvs simulator | |
| ■ Sample CDF parameters for auLvs. | |


© 2023 Cadence Design Systems, Inc.

# 19

# Working with UltraSim Simulator

This chapter covers the following topics to describe the UltraSim simulator interface and simulator options:


- Specifying UltraSim Simulator Interface on page 1208
- Specifying UltraSim Options on page 1210
- Saving Outputs for UltraSim Simulations on page 1230
- UltraSim Power Network Solver on page 1224
- UltraSim Interactive Simulation Debugging on page 1228



© 2023 Cadence Design Systems, Inc.

# Specifying UltraSim Simulator Interface

You can run Virtuoso UltraSim 64-bit software by specifying one of the following options:

1. Use the `-debug3264 -V` command to check your system configuration:

   `$`*`your_install_dir`*`/tools/bin/ultrasim -debug3264 -V`

   You can use the information provided by the command to verify if the 64-bit version is applicable to your platform, if the 64-bit software is installed, and whether or not it is selected.

2. Install the Virtuoso UltraSim 64-bit software to the same location as your 32-bit software.

3. Verify that all required software patches are installed by running `checkSysConf` (system configuration checking tool script). The script is located in your local installation of Cadence software:

   `$`*`your_install_dir`*`/tools/bin/checkSysConf MMSIM6.0`

   The script is also available on Cadence Online Support, customer support system.

4. Set the `CDS_AUTO_64BIT` environment variable `{ALL|NONE|"list"|INCLUDE:"list"|EXCLUDE:"list"}` to select 64-bit executable.

   - **`ALL`** invokes all applications as 64-bit.

     The list of available executable files is located at:

     `$`*`your_install_dir`*`/tools/bin/64bit`

   - **`NONE`** invokes all applications as 32-bit.

   - **`"list"`** invokes only the executable files included in the list as 64-bit.

     `"list"` is a list of case-sensitive executable names delimited by a comma (`,`), semicolon (`;`), or colon (`:`).

   - **`INCLUDE:"list"`** invokes all applications in the list as 64-bit.

   - **`EXCLUDE:"list"`** invokes all applications as 64-bit, except the applications contained in the list.

   **Note:** If `CDS_AUTO_64BIT` is not set, the 32-bit executable is invoked by default.

   **Example**

   ```
   setenv CDS_AUTO_64BIT ultrasim
   setenv CDS_AUTO_64BIT "EXCLUDE:si"
   ```

5. Launch the executable files through the wrapper.


© 2023 Cadence Design Systems, Inc.

All 64-bit executables are controlled by a wrapper executable. The wrapper invokes the 32-bit or 64-bit executables depending on how the `CDS_AUTO_64BIT` environment variable is set, or whether the 64-bit executable is installed. The wrapper also adjusts the paths before invoking the 32-bit or 64-bit executable files. The wrapper you use to launch the executables is located at *your_install_dir/tools/bin.*

**Note:** Do not launch the executables directly from the *your_install_dir/tools/bin/64bit* or *your_install_dir/tools/bin/32bit* directory.

**Example**

*$your_install_dir*`/tools/bin/ultrasim`

6. In the Explorer window, choose *Setup – Simulator*. The *Choosing Simulator -- ADE Explorer* form appears. Choose simulator as `UltraSim` in this form.

For more information about setting Virtuoso UltraSim simulator options, refer to the *Virtuoso UltraSim Simulator User Guide*.

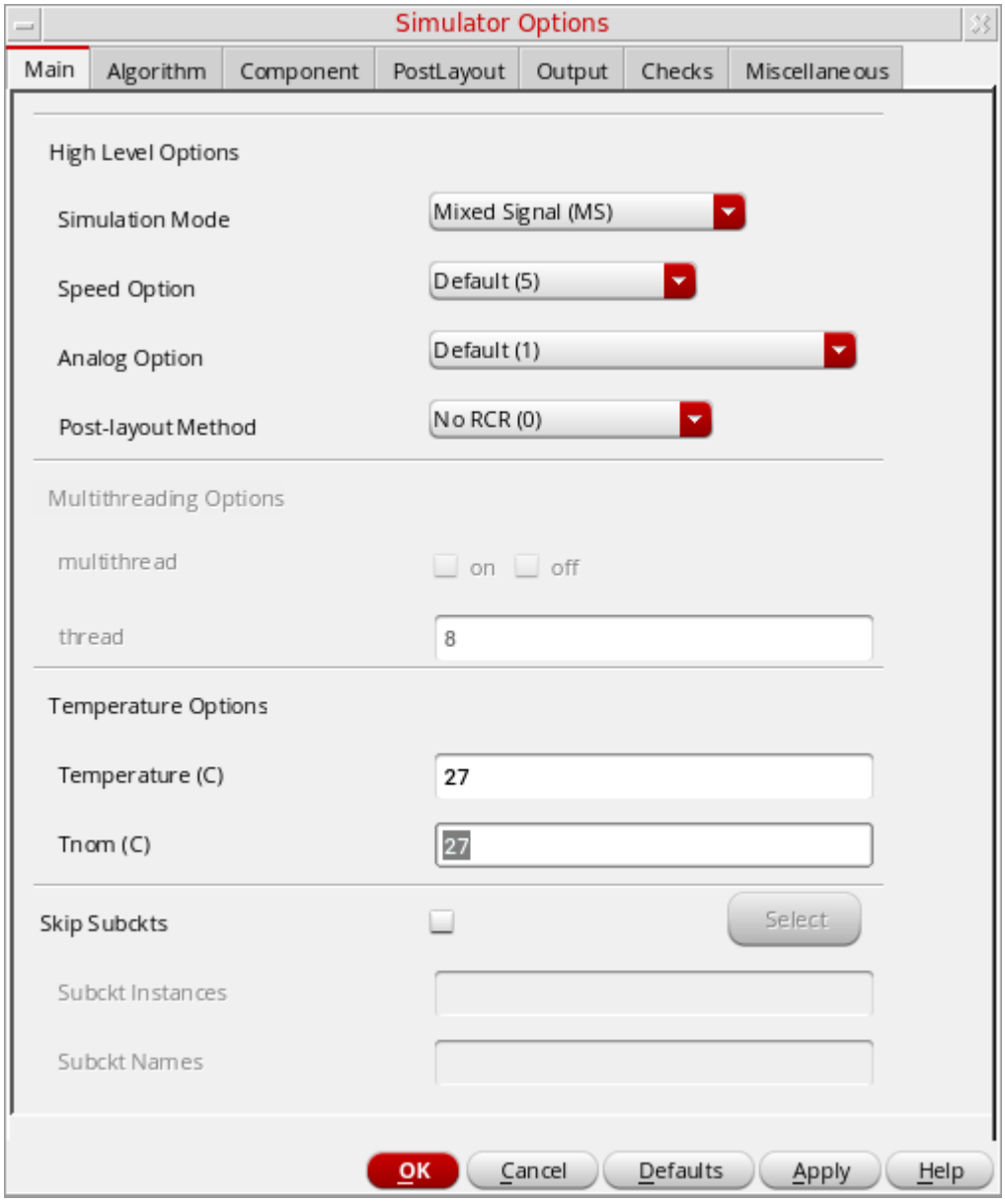

© 2023 Cadence Design Systems, Inc.

# Specifying UltraSim Options

To set the Virtuoso UltraSim simulator options,

1. In the Explorer window, choose *Simulation – Options – Analog.*

   The *Simulator Options* form appears.




2. Set the simulator options as needed.


© 2023 Cadence Design Systems, Inc.

For more details about the Virtuoso UltraSim simulator options, refer to the *Virtuoso UltraSim Simulator User Guide*.

3. Click *OK*.

### Setting Voltage Regulator Simulation Options

The voltage regulator (VR) simulation feature in the UltraSim simulator allows you to simulate designs with large circuit blocks powered by internal voltage regulators. For more information about VR simulation, see the *Voltage Regulator Simulation* in the *Virtuoso UltraSim Simulator User Guide*.

To set the voltage regulator options,

1. In the Explorer window, choose *Simulation – Options – Analog*.

   The Simulator Options form appears.

2. Click the *Main* tab.

3. Ensure that the global simulation mode specified in the *Simulation Mode* drop-down field is `Digital Extended (DX)`, `Digital Fast (DF)`, `Mixed Signal (MS)` or `Digital Accurate (DA)`.

   **Note:** VR simulation can be enabled only if the global simulation mode is Digital Extended (DX), Digital Fast (DF), Mixed Signal (MS) or Digital Accurate (DA).

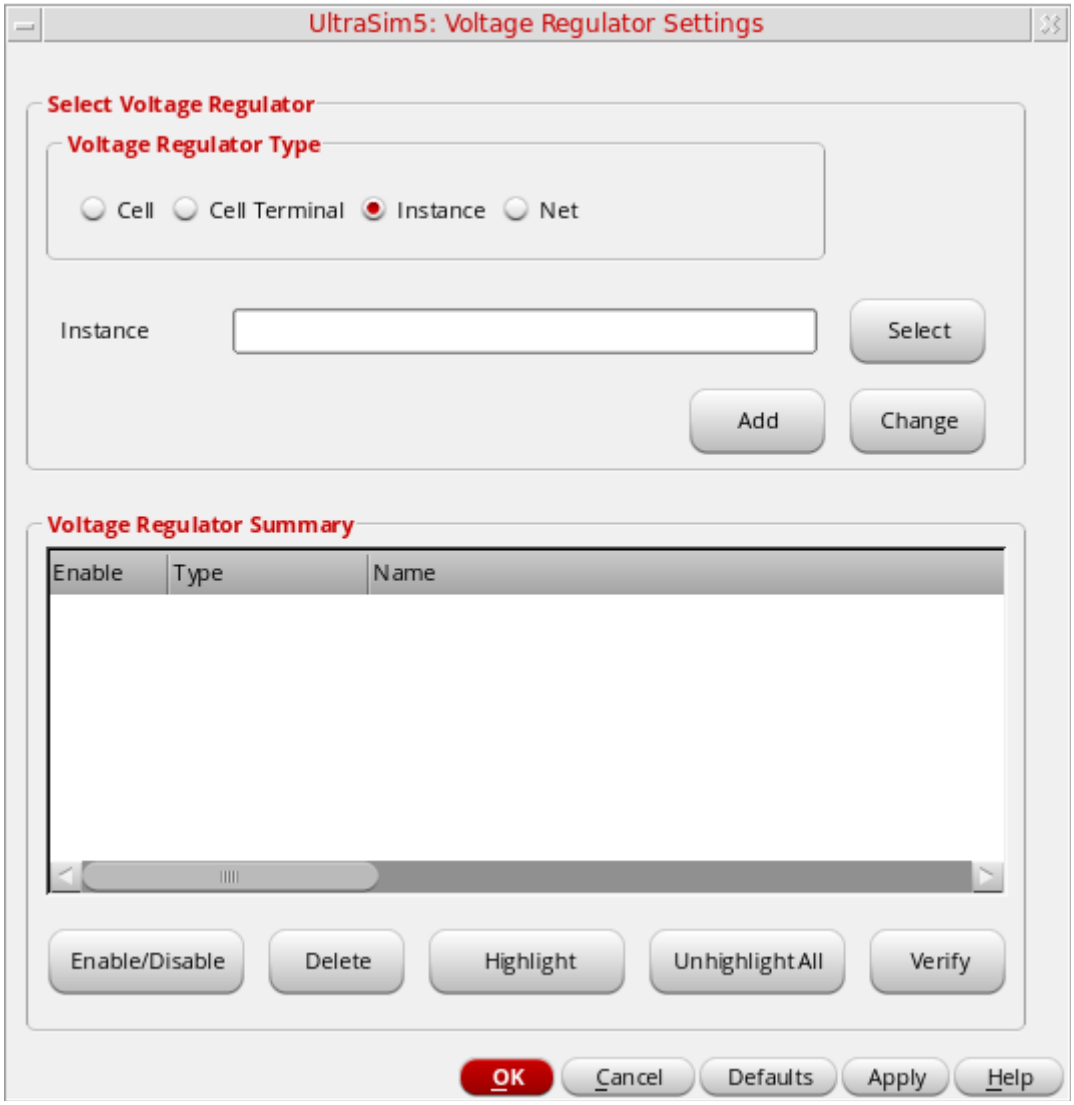
4. Click the  tab.

5. Select the *Voltage Regulator* check box to enable VR simulation.

6. Click the *Setting* button next to the *Voltage Regulator* check box.


© 2023 Cadence Design Systems, Inc.

The Voltage Regulator Settings form appears.




You can use this form to specify the cells, cell terminals, instances and nets to be used for VR simulation. Note the following:

**Note:** You can specify cell terminals only if you are using the simulation front end parser (SFE) with the Ultrasim simulator. For information about using the SFE parser with the Ultrasim simulator, see the *Spectre AMS Designer and Xcelium Simulator Mixed-Signal User Guide*.

For more information, see the following topics:

- Adding Objects for VR Simulation on page 1213
- Verifying the Objects Added for VR Simulation on page 1214


© 2023 Cadence Design Systems, Inc.

- Disabling and Enabling Objects for VR Simulation on page 1215
- Modifying Objects on page 1215
- Deleting Objects on page 1215
- Highlighting and Dehighlighting Enabled Objects on the Schematic on page 1215

### *Adding Objects for VR Simulation*

To add an object (cell, cell terminal, instance or net) for VR simulation,

1. Select the voltage regulator type.

| Select | To |
|---|---|
| Cell | Add a voltage regulator cell. |
| Cell Terminal | Add a cell terminal.<br>The terminal must be an output voltage terminal delivering supply voltages to digital cells or channel connected devices. |
| Instance | Add a voltage regulator instance, or an output device of the voltage regulator in case your circuit has no hierarchy. |
| Net | Add a net.<br>The net must be an output voltage terminal delivering supply voltages to digital cells or channel connected devices. |

2. Enter the name of the object in the text field, or click the *Select* button to select the object on the schematic.
   - If the voltage regulator type is *Cell*, enter the cell name in the *Cell* field, or click the *Select* button to select the cell on the schematic.
   - If the voltage regulator type is *Cell Terminal*, enter the cell terminal name in the *Cell Terminal* field using the format `"library" "cell" "port"`, or click the *Select* button to select the cell terminal on the schematic.
   - If the voltage regulator type is *Instance*, enter the instance name in the *Instance* field, or click the *Select* button to select the instance on the schematic.


© 2023 Cadence Design Systems, Inc.

- If the voltage regulator type is *Net*, enter the net name in the *Net* field, or click the *Select* button to select the net on the schematic.

3. Click *Add*.

   The object is displayed in the Voltage Regulator Summary table.

   **Note:** By default, the object is enabled for VR simulation. For more information, see Disabling and Enabling Objects for VR Simulation on page 1215.

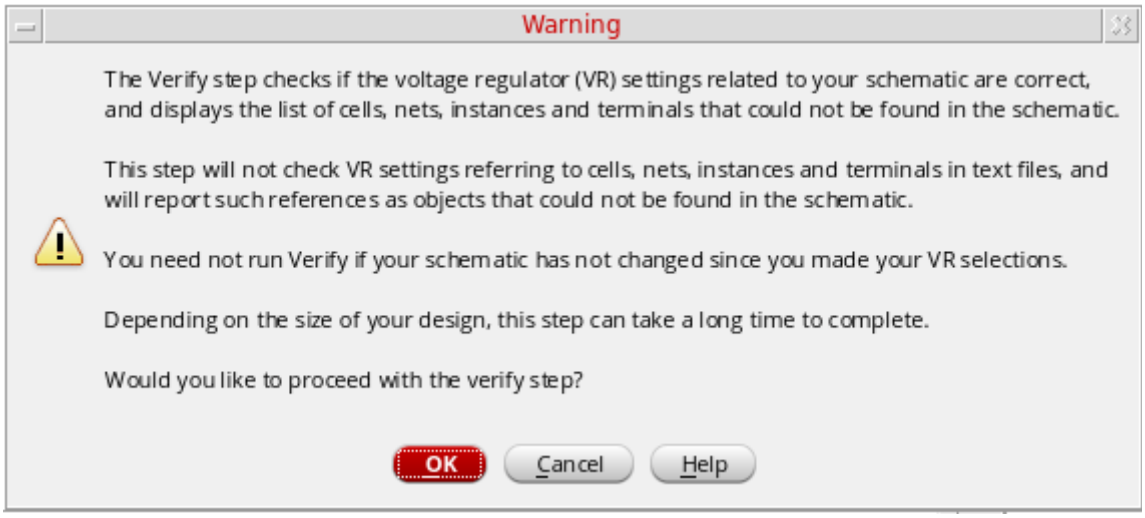
### *Verifying the Objects Added for VR Simulation*

After specifying the objects for VR simulation, you can verify whether the objects exist in the schematic.

To verify whether the objects enabled for VR simulation exist in the schematic,

1. Click the *Verify* button.

   The following message box appears:




2. Click *OK*.

If any of the objects do not exist in the schematic, a message box appears displaying the list of such objects.

Note the following:

- Only the objects that are enabled in the Voltage Regulator Summary table are verified.


© 2023 Cadence Design Systems, Inc.

- The verification process does not check objects that exist in text files, and will report such objects as not found in your schematic.

### *Disabling and Enabling Objects for VR Simulation*

You can disable and enable objects for VR simulation. The *On* status in the *Enable* column in the Voltage Regulator Summary table indicates that the object is enabled for VR simulation. The *Off* status in the *Enable* column indicates that the object is disabled for VR simulation.

To disable an object for VR simulation,

➤ Select the object in the Voltage Regulator Summary table and click the *Enable/Disable* button.

To enable an object for VR simulation,

➤ Select the object in the Voltage Regulator Summary table and click the *Enable/Disable* button.

### *Modifying Objects*

To modify an object in the Voltage Regulator Settings form,

1. Select the object in the Voltage Regulator Summary table.

   The information for the object is displayed in the *Select Voltage Regulator* group box.

2. Modify the object information and click the *Change* button.

   The changes are displayed in the Voltage Regulator Summary table.

### *Deleting Objects*

To delete an object in the Voltage Regulator Settings form,

➤ Select the object in the Voltage Regulator Summary table and click the *Delete* button.

### *Highlighting and Dehighlighting Enabled Objects on the Schematic*

The objects that are enabled for VR simulation can be highlighted on the schematic.

To highlight an enabled object on the schematic,

➤ Select the object in the Voltage Regulator Summary table and click the *Highlight* button.


© 2023 Cadence Design Systems, Inc.

The selected objects are highlighted in yellow color on the schematic. If the selected object is a cell, all instances of the cell are highlighted on the schematic.

**Note:** To highlight multiple objects, hold down the *Shift* key (for contiguous selection) or the *Ctrl key* (for noncontiguous selection) and click the next object to add more objects to the selection set, then click the *Highlight* button.

To dehighlight all objects on the schematic,

➤ Click the *Unhighlight All* button.

### Setting Skip Options in Ultrasim

To set the Skip options in Ultrasim:

1. In the Explorer window, choose *Simulation – Options – Analog*.
2. The Simulator Options form appears, choose *Main Tab*.

The *Skip Subckts* options appears in the *Main* Tab of Simulator Options form. Set the skip options as required.

You can enter multiple subcircuit instances and names in respective fields. You can also directly select instances from the schematic by using the *Select* button.

### Setting Block-Based usim_opt Options

#### *Schematic*

You can use the block-based *usim_opt on schematics* option to set speed, accuracy, and functionality of the Virtuoso UltraSim simulation for local subcircuit instances. Any Virtuoso UltraSim simulator options, including *usim_opt*, can be set in instance properties on the schematic. The most commonly used options are

- `sim_mode` (df/da/ms/a/s)
- `speed` (1-8)


© 2023 Cadence Design Systems, Inc.

- `analog` (1/2/3)

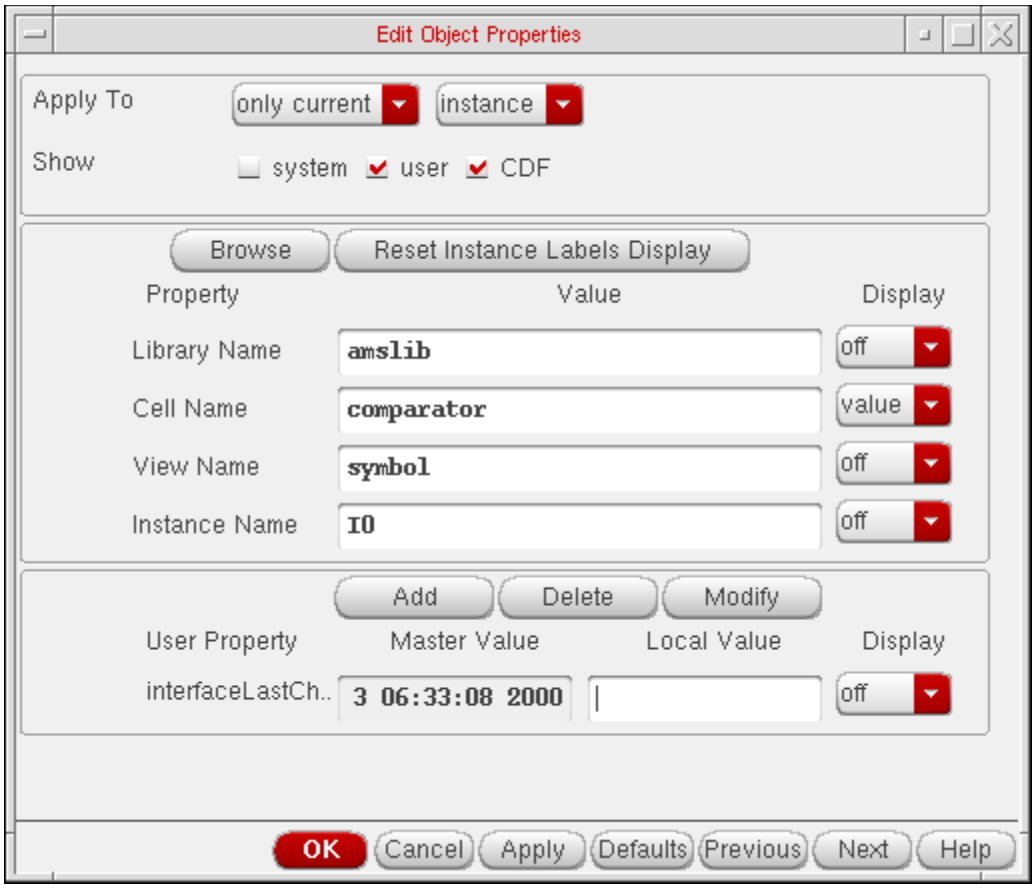
**Note:** The *usim_opt* option is valid only for block-level (subcircuit instances) settings.

For more details, refer to *Simulation Options* chapter in the *Virtuoso UltraSim Simulator User Guide*.

To set block-based *usim_opt on schematic* options

1. Click an instance in the schematic.
2. In the schematic window, choose *Edit – Properties – Objects*.

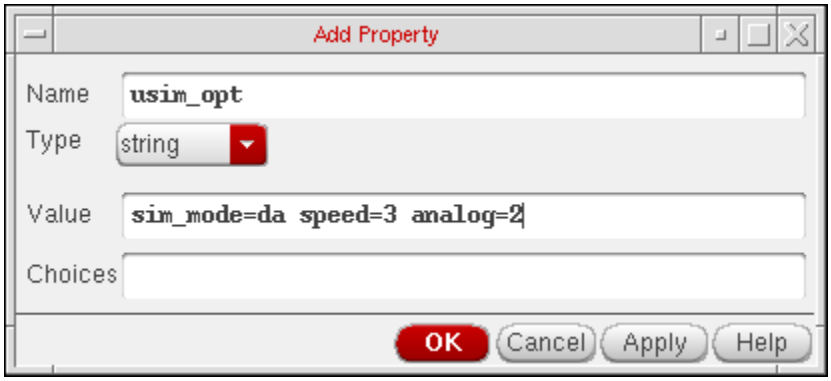
   The Edit Object Properties form appears.




3. Click *Add*.


© 2023 Cadence Design Systems, Inc.

The Add Property form appears.




4. In the *Name* field, type `usim_opt`.
5. Type `sim_mode=da speed=3 analog=2` into the *Value* field.
6. Click *OK* to save the settings and close the Add Property form.
7. In the Edit Object Properties form, click *OK*.
8. In the schematic, choose *Design – Check and Save*.
9. In the Cadence® Analog Design Environment simulation window, choose *Simulation – Options – Analog*.

   The Simulator Options form appears.
10. Turn on *Allow usim_opt on schematics*.
11. Click *OK* to save the settings and close the Simulator Options form.
12. Run netlisting or the simulation from the Simulation window.

    The *usim_opt* settings are set locally for the instance block in the netlist.

    **Note:** Spectre and HSPICE netlist formats are supported in IC 5.0 and later releases.

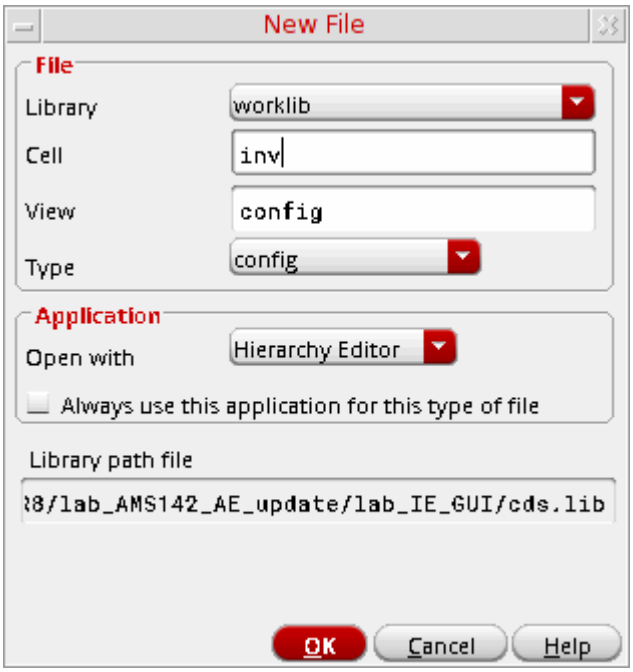
### *Hierarchy Editor*

You can also set block level *usim_opt* options, such as speed and accuracy, using the Cadence® Hierarchy Editor (HED). For more information about setting Virtuoso UltraSim simulator options, refer to *Simulation Options* chapter in the *Virtuoso UltraSim Simulator User Guide*.

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


© 2023 Cadence Design Systems, Inc.

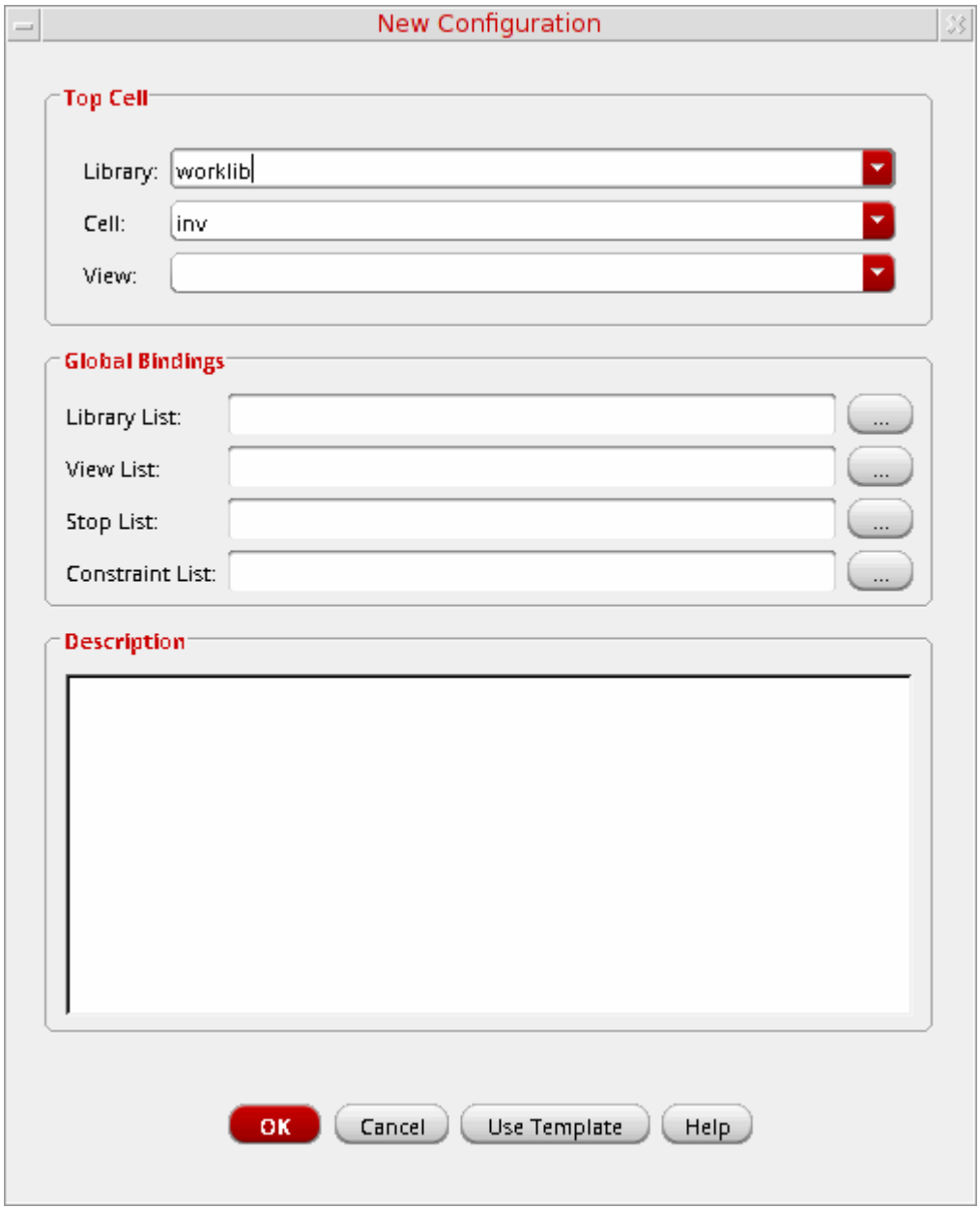
The Create New File form appears.




2. Choose a library, cell, and view.
3. Choose *Hierarchy-Editor* from the *Tool* drop-down list box.
4. Click *OK*.


© 2023 Cadence Design Systems, Inc.

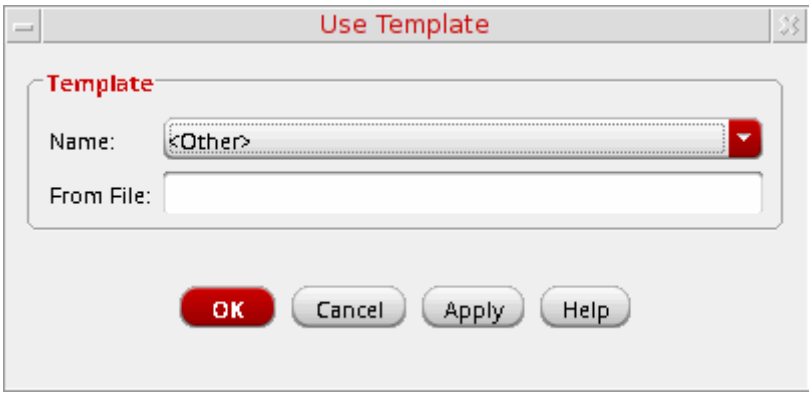
The New Configuration form appears.




5. Click the *Use Template* button located at the bottom of the form.


© 2023 Cadence Design Systems, Inc.

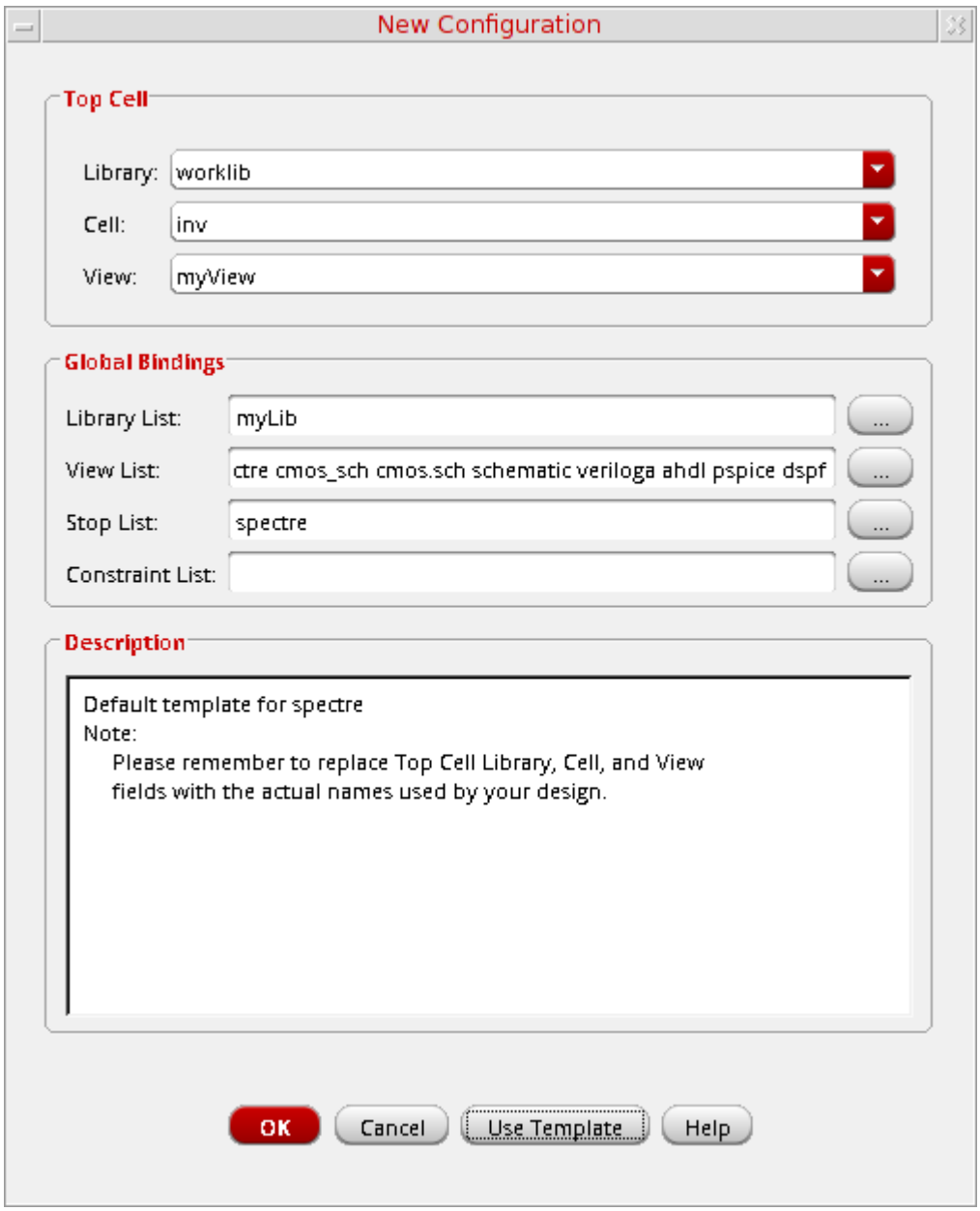
The Use Template form appears.




6. Choose *spectre* from the *Name* drop-down list box.
7. Click *OK*.


© 2023 Cadence Design Systems, Inc.

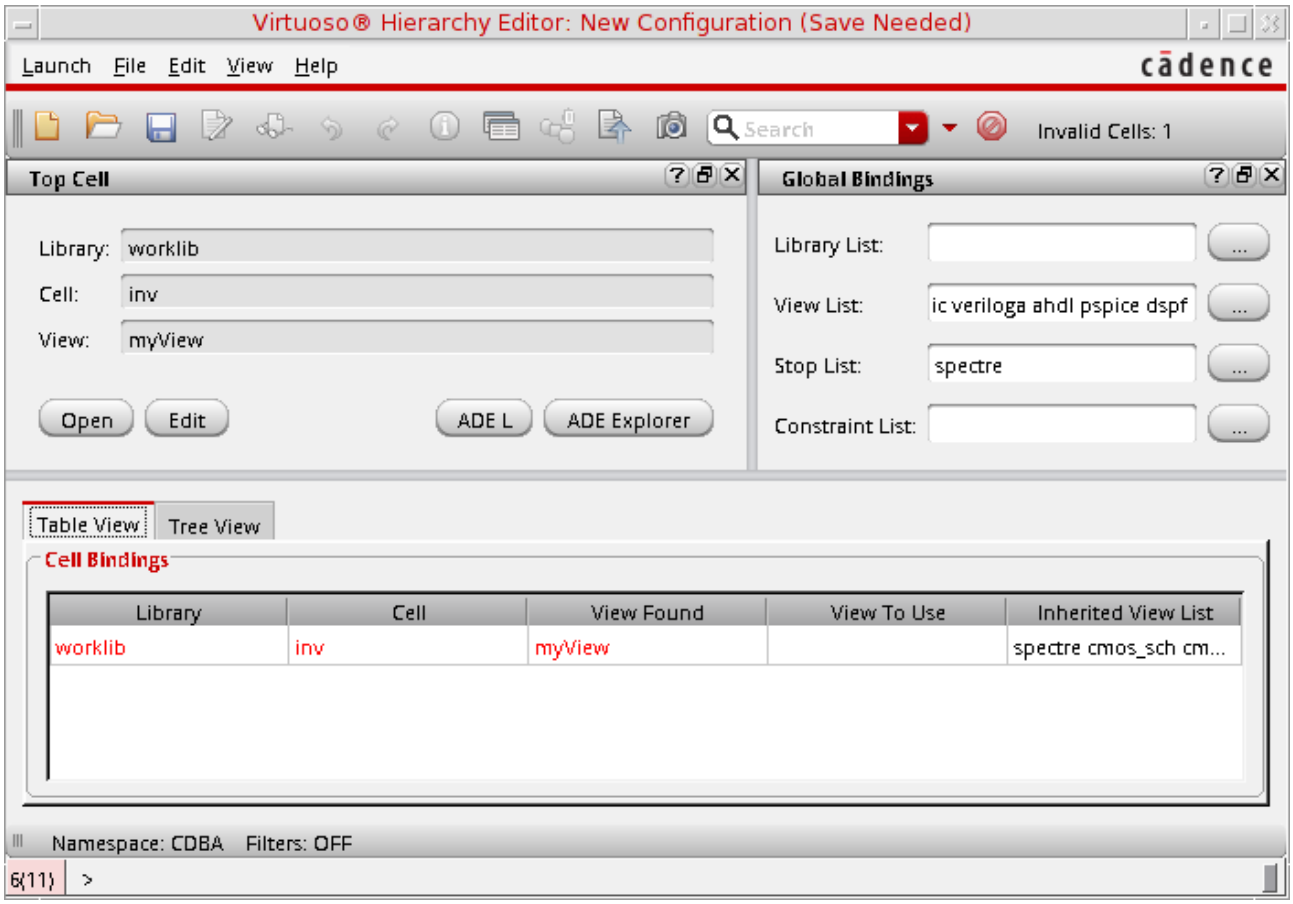
The New Configuration form redisplays with default data for the *Top Cell* and *Global Bindings* sections.




8. In the *Top Cell* section, enter the desired library, cell name, and schematic view.
9. In the *Global Bindings* section, remove `myLib` from the *Library List* field.
10. Click *OK*.


© 2023 Cadence Design Systems, Inc.

The Cadence hierarchy editor form displays your data.




**Note:** The hierarchy editor form configures the design by using a default *View List* and *Stop List* in the *Global Bindings* section. You need to modify these lists for your design.

**11.** Choose *View – Properties*.

The *sim_mode* and *speed* columns appear in the *Cell Bindings* section.

**12.** Choose *Edit – Add Property Column* to add additional cell- or occurrence-based properties to the *Cell Bindings* section.

**Note:** Instance-based properties are not supported by the Virtuoso UltraSim simulator in Virtuoso ADE Explorer.

**13.** Set options for cell- or occurrence-based properties.

To set options for cell-based properties,

**a.** Right-click in a property column.


© 2023 Cadence Design Systems, Inc.

A popup menu appears.

**b.** Choose *Set "property name" Cell Property* to set the property.

To set options for occurrence-based properties,

**a.** Choose *View – Tree*.

**b.** Select an instance and right-click a property column in the *Cell Bindings* section.

A popup menu appears.

**Note:** Do not left-click a property column to set a property (left-click creates an instance-based property, which is not supported by the Virtuoso UltraSim simulator in ADE).

**c.** Choose *Set "name" Occurrence Property*.

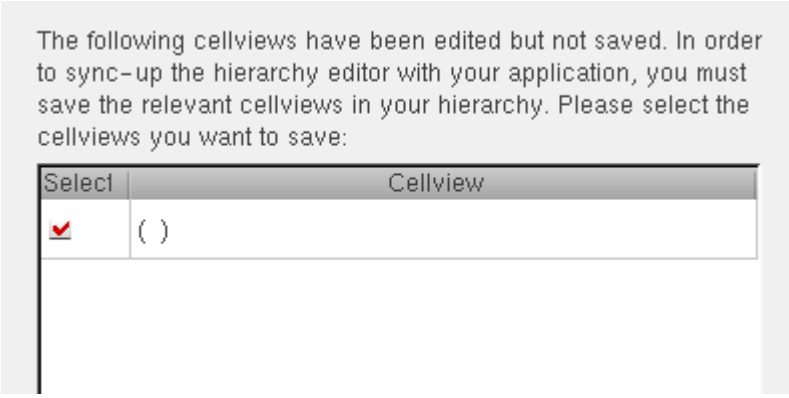
**d.** Choose the appropriate property value.

**e.** Click *OK*.

The instance is marked with an ○ symbol.

**f.** Choose *View – Update*.

The Update Sync-up form appears.




**g.** Click *OK* to save the cellview.

# UltraSim Power Network Solver

This section describes how to detect and analyze power networks using the Virtuoso® UltraSim™ power network solver (UPS) in the analog design environment.

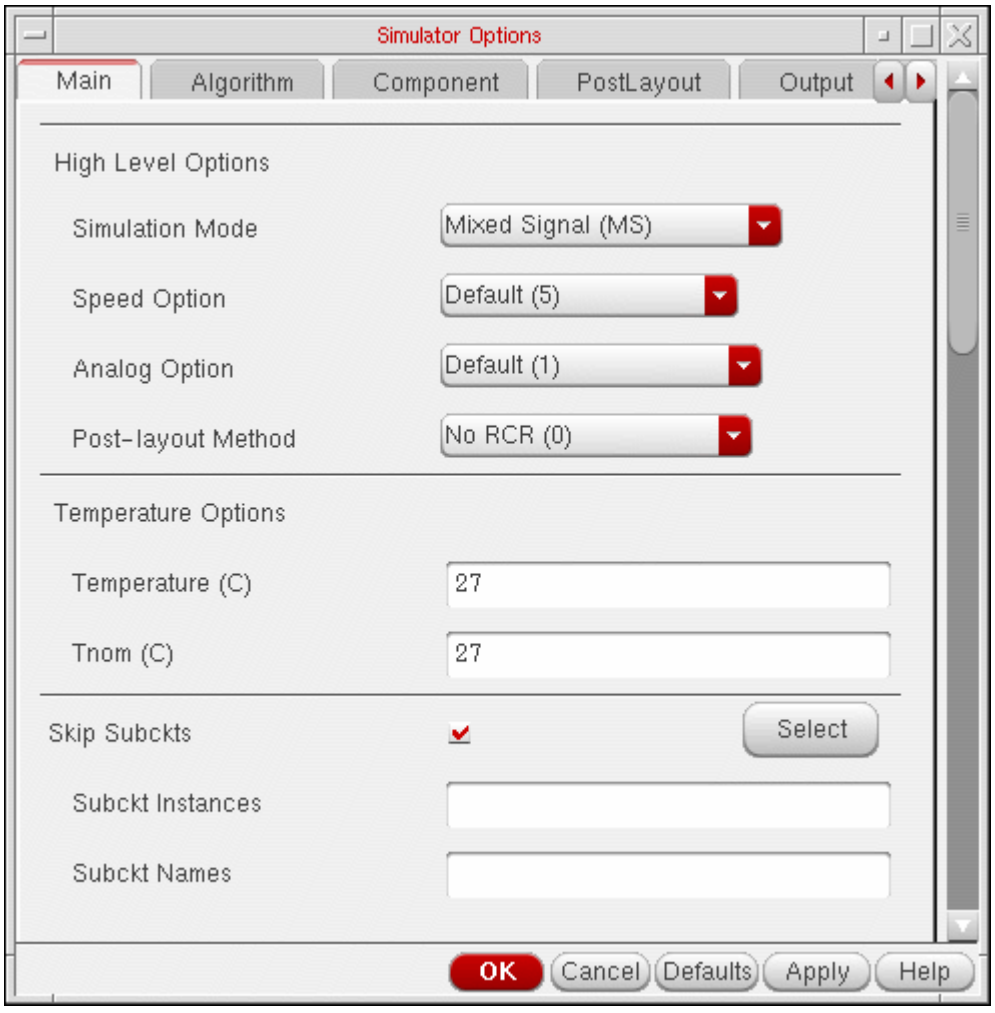

© 2023 Cadence Design Systems, Inc.

To analyze IR drop effects and their influences on circuit behavior, parasitics in the power and ground net of a circuit design need to be extracted and analyzed together with the circuit. Parasitic elements, such as resistors, capacitors, and inductors, build the power network. These elements need to be simulated, so the parasitic effects on circuit behavior can be analyzed.

To use UPS to detect and analyze power networks:

1. Choose *Simulation – Options – Analog*.

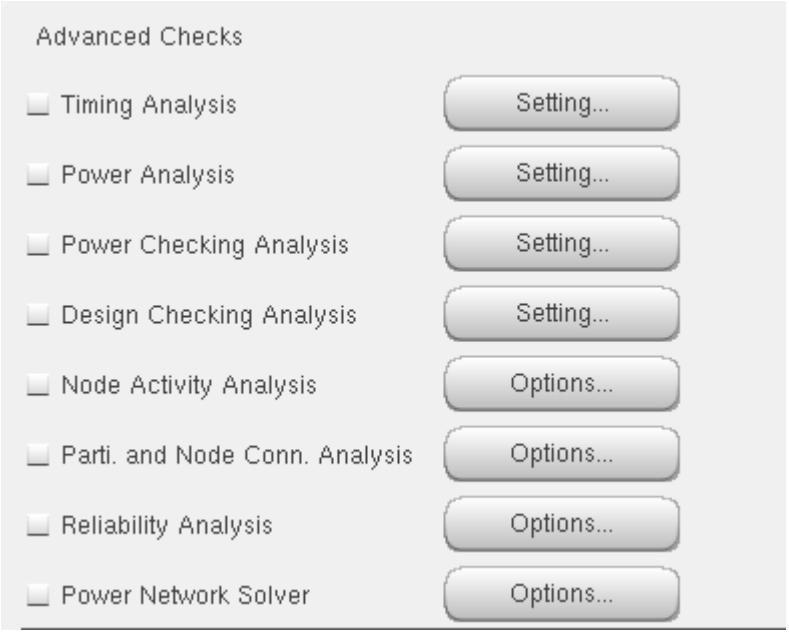
   The Simulator Options form appears.





© 2023 Cadence Design Systems, Inc.

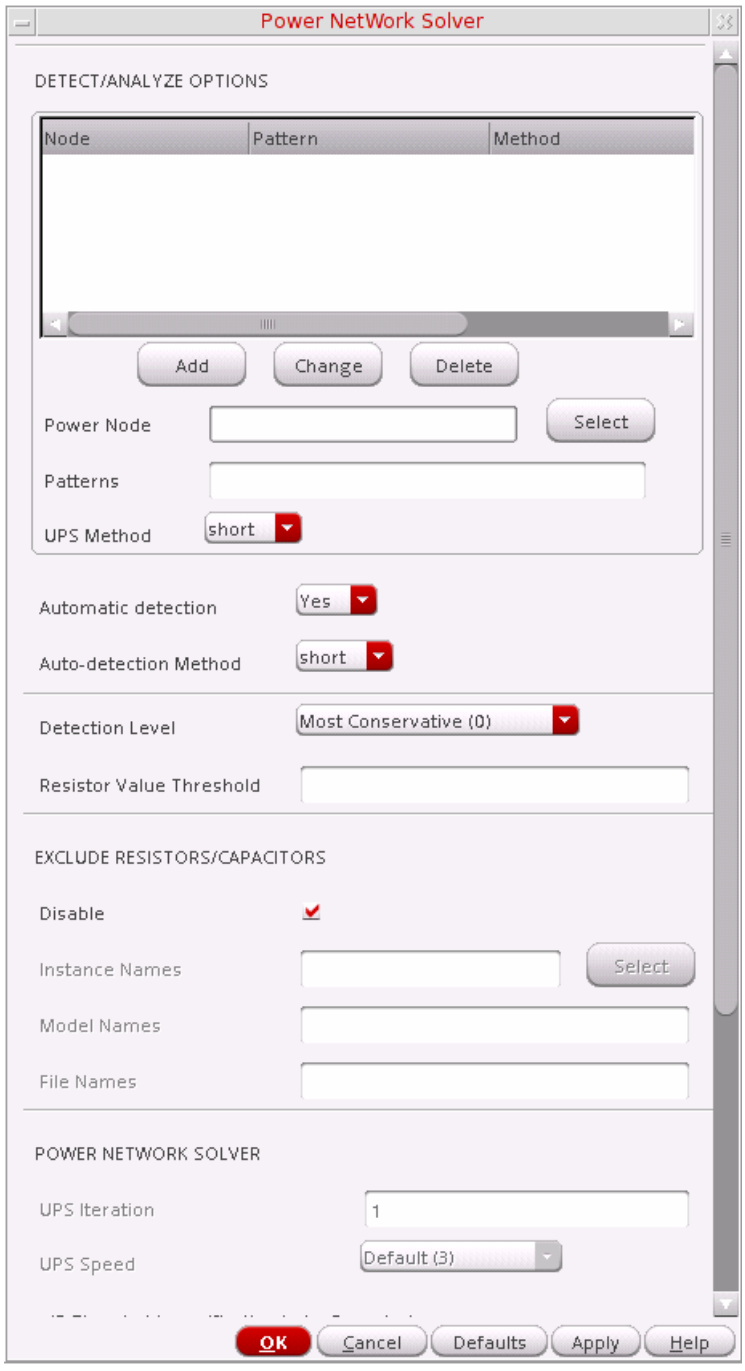
2. Click the *Power Network Solver* check box in the *Checks* Tab.




3. Click the *Options* button.


© 2023 Cadence Design Systems, Inc.

The Power Network Solver window appears.





© 2023 Cadence Design Systems, Inc.

4. Adjust the power network solver options as needed.
5. Click *OK*.

For more information, refer to Chapter 5, "Power Network Solver" in the *Virtuoso UltraSim Simulator User Guide*.

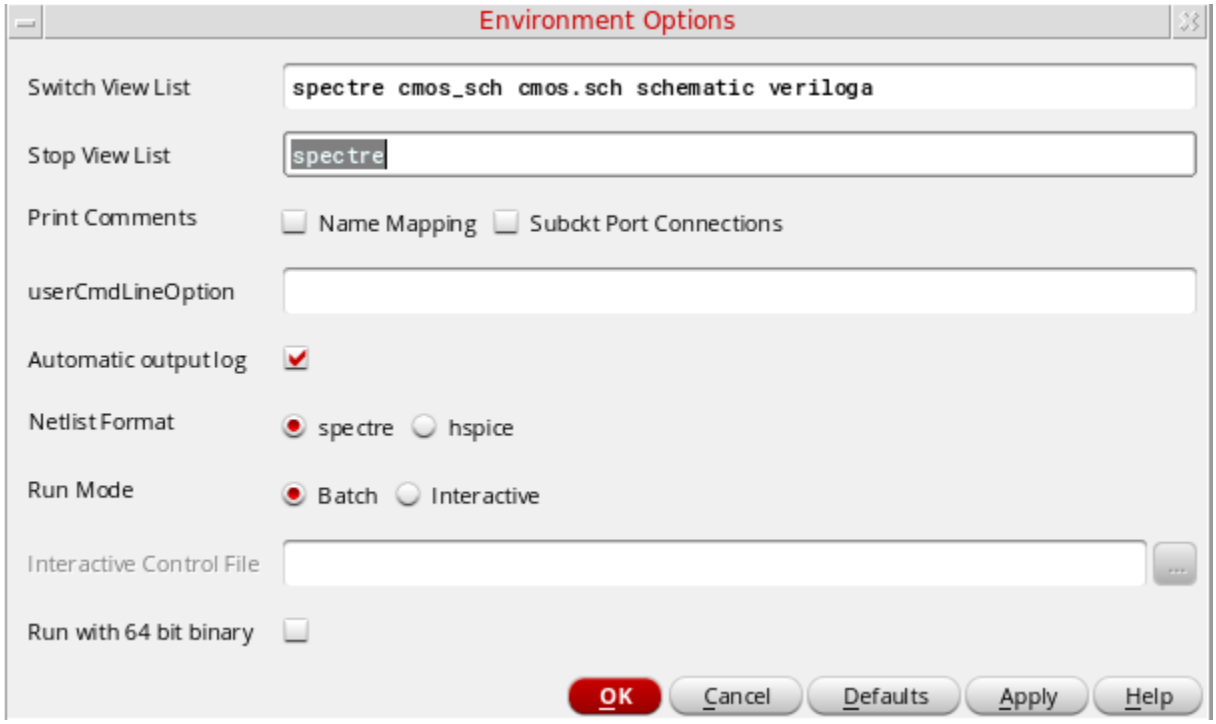
# UltraSim Interactive Simulation Debugging

The Virtuoso UltraSim simulator interactive circuit debugging mode allows you to obtain design data, such as circuit elements and parameters, circuit topology, and instantaneous signal values. It can also be used to probe dynamic circuit behavior, including voltage and current waveforms simulated to the current time step.

To use the interactive simulation debugging mode:

1. Choose *Setup – Environment*.

   The Environment Options window appears.




2. In the Run Mode field, select *Interactive*.
3. Type in the *Interactive Control file*name.


© 2023 Cadence Design Systems, Inc.

4. Click *OK.*

For more information about interactive debugging, refer to Chapter 6, “Interactive Simulation Debugging” in the *Virtuoso UltraSim Simulator User Guide*.

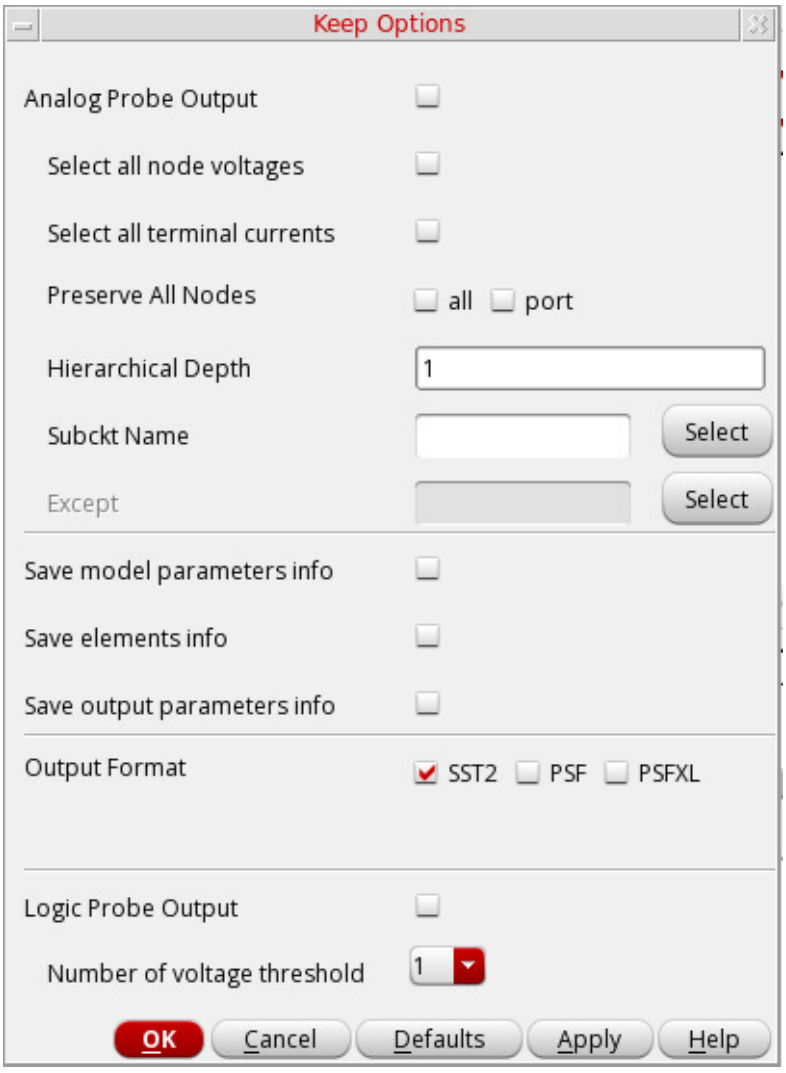

© 2023 Cadence Design Systems, Inc.

# Saving Outputs for UltraSim Simulations

To view outputs for waveform data:

1. In the ADE Explorer window, choose *Outputs – Save All*.

   The *Keep Options* form appears.




### Analog Probe Output

To output waveform data for an analog probe, choose the appropriate settings.

- **Select all node voltages** use to output all node voltages.


© 2023 Cadence Design Systems, Inc.

- **Select all terminal currents** use to output all terminal currents. The Virtuoso UltraSim simulator outputs the first terminal current of each device.
- **Preserve All Nodes** use to preserve either all or port RC node voltages. RC nodes are not reduced, allowing the nodes to be saved in simulation.
- **Hierarchical Depth** use to save and display more than one level of hierarchical results.
- **Subckt Name** use to indicate the subcircuit name for the analog probe. If a name is not entered into the `subckts` field, the simulator applies the analog probe to all blocks.
- **Except** specifies the nodes to be excluded from the analog probe. A node name, element name, or wildcard (`*`) can be used.

**Save model parameters info** specifies that input parameters for models of all components be saved.

**Save elements info** specifies that input parameters for instances of all components be saved.

**Save output parameters info** specifies that effective and temperature-dependent parameter values be saved.

**Output Format** specifies the format in which the results data must be saved. The results data can be saved in the following formats:

- **SST2** – Signal Scan 2 format. This format is supported for transient analyses only. Non-transient data cannot be generated in the SST2 format and is generated in parameter storage format (PSF) format.
- **PSF** – Cadence parameter storage format (PSF)

  **Note:** The UltraSim simulator always uses PSF version 1.0 for writing the simulation results.
- **PSFXL** – Cadence parameter storage XL format that provides higher performance for large circuit designs.

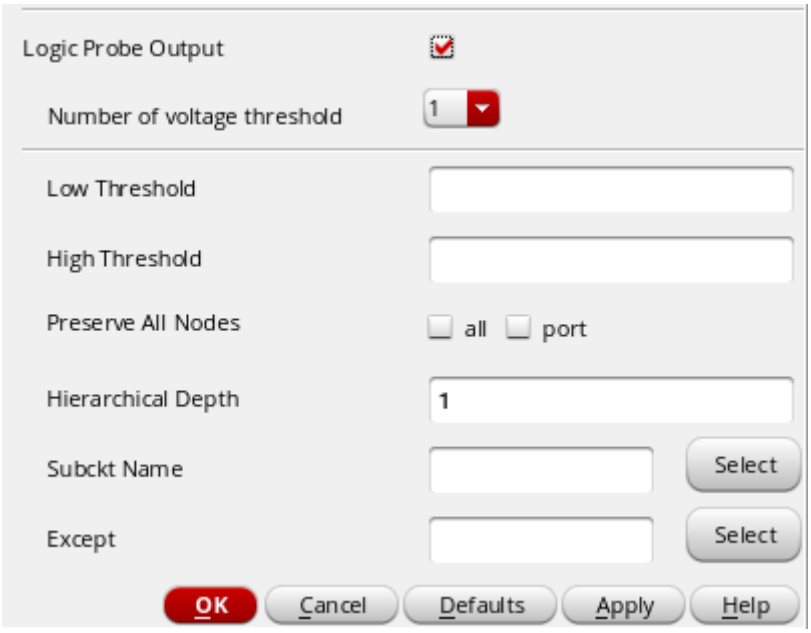
***The PSFXL format is not compatible with versions of Virtuoso Visualization and Analysis prior to IC6.1.4 unless the CDS_PSFXL_COMPAT environment variable is defined by the user.***

**Logic Probe Output**


© 2023 Cadence Design Systems, Inc.

To output waveform data for a logic probe, choose the appropriate settings (the Keep Options form expands to show the *Logic Probe Output* settings).

**Note:** The *Logic Probe Output* option is available only for SignalScan 2 (SST2) and fast signal database (FSDB) waveform output formats.




- **Number of voltage threshold** use to indicate the number of voltage thresholds for each logic probe.
- **Low Threshold** specify for each logic probe the low threshold value which corresponds to the digital `0` value.
- **High Threshold** specify for each logic probe the high threshold value which corresponds to the digital `1` value.
- **Preserve All Nodes** use to preserve all or only port RC nodes from the RC reduction.
- **Hierarchical Depth** use to save and display more than one level of hierarchical results.
- **Subckt Name** use to indicate the subcircuit name for the logic probe. If a name is not entered into the `subckts` field, the simulator applies the logic probe to all blocks
- **Except** specifies the nodes to be excluded from the logic probe. A node name, element name, or wildcard (`*`) can be used.

2. Click *OK*.


© 2023 Cadence Design Systems, Inc.

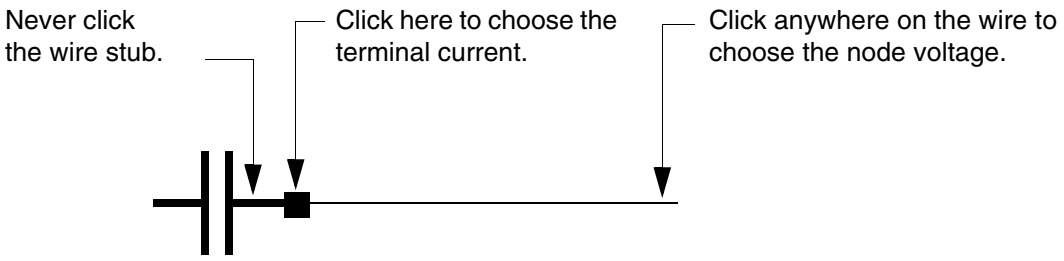
## Saving Selected Voltages or Currents

To save the simulation data for particular nodes and terminals,

1. In the Simulation window, choose *Outputs – To Be Saved – Select On Design*.
2. In the Schematic window, choose one or more nodes or terminals.

   The system circles pins when you choose a current and highlights wires when you choose a net.

   - Click an instance to choose all instance terminals.
   - Click the square pin symbols to choose currents.
   - Click wires to choose voltages.
   - Click and drag to choose voltages by area.




Press the `Esc` key when you finish


© 2023 Cadence Design Systems, Inc.

© 2023 Cadence Design Systems, Inc.

# A

# auCdl Netlisting

This appendix describes the auCdl (Analog and Microwave Circuit Description Language) netlisting procedure. It contains details on parameters required for auCdl and also the different ways to netlist to auCdl. This information is applicable to any 4.4 version of the Virtuoso® design framework II (DFII).

This appendix covers the following topics:

- What Is auCdl and Why Do You Need It? on page 1235
- Licensing Requirements
- Running auCdl on page 1236
- Customization Using the .simrc File on page 1244
- Support for HED Features on page 1256
- Custom Netlisting Procedures on page 1257
- Blackbox Netlisting on page 1270
- Additional Customizations on page 1274
- CDF Simulation Information for auCdl on page 1294
- Complete Example on page 1301

## What Is auCdl and Why Do You Need It?

To compare a layout versus a schematic (LVS) using LOGLVS, you need a netlist representation of the schematic for a design for LOGLVS. The netlist must be in CDL (Circuit Description Language) format. To create a CDL netlist for an analog circuit, you use a netlister called auCdl (Analog and Microwave Circuit Description Language).


© 2023 Cadence Design Systems, Inc.

# Licensing Requirements

You must have one of the following licenses to run auCdl from DFII or from the command-line. If one of these licenses are not already checked out, the first available license will be checked out in the following order when you run auCdl:

- 95100 Virtuoso® Schematic Editor L
- 95115 Virtuoso® Schematic Editor XL
- 206 Virtuoso® Simulation Environment

# Running auCdl

You can run auCdl from within or outside the DFII environment.

To translate files from the DFII database format into an auCdl netlist,

1. Set the `CDS_Netlisting_Mode` environment variable as given below:

   ```
   setenv CDS_Netlisting_Mode "Analog"
   ```

2. Create an auCdl view for the cell. For more information, see Creating a config view for auCdl on page 1240.
3. Add the auCdl simulation information to the cell's CDF. For more information, see CDF Simulation Information for auCdl on page 1294.

You can customize the auCdl Netlister using the simulation run control (`.simrc`) file. For more information, see Customization Using the .simrc File on page 1244.

## Running auCdl from within DFII

In DFII, you can extract the auCdl netlist by doing the following:

1. In the CIW, choose *File – Export – CDL*.
2. In the CDL Out form, fill in the appropriate fields and click *OK* or *Apply*.

For more information about using CDL Out, read the *Translating CDL Files* section in the *Design Data Translator's Reference*.


© 2023 Cadence Design Systems, Inc.

## Running auCdl from the Command-Line

To run CDL Out from the command-line, you must create a simulation environment (`si.env`) file in advance and name the file as a command argument. Run CDL Out interactively once to create the `si.env` file. Once the `si.env` file is created,

1. Copy the `cds.lib` file to the run directory.
2. Enter the following command:

   ```
   si -batch -command netlist
   ```

   the `-batch` option runs CDL in batch mode and the `-command netlist` option generates an ASCII netlist file.

CDL Out can generate a hierarchical netlist. CDL Out generates a netlist hierarchy that duplicates the hierarchy of your design. Each cell in your schematic becomes a separate subcircuit in the netlist. The hierarchical netlister automatically prefixes each instance name with the proper character for its element type; for example, `"M"` for MOSFET and `"R"` for resistor. This prefixing minimizes mapping and name translation.

### The si.env File

The following is an example of a `si.env` file followed by description of each of these properties.

```
simLibName = "testLib"
simCellName = "testTop"
simViewName = "schematic"
simSimulator = "auCdl"
simNotIncremental = nilsimReNetlistAll = nil
simViewList = '("auCdl" "schematic" "gate.sch" "cmos.sch")
simStopList = '("auCdl")
simNetlistHier = t
hnlNetlistFileName = "netlist"
simRunDir = "/cds/1.0/test/translator/cdlout/paramCase/"
resistorModel = " "
shortRES = 2000.0
preserveRES = 'nil
checkRESVAL = 'nil
checkRESSIZE = 'nil
preserveCAP = 'nil
checkCAPVAL = 'nil
checkCAPAREA = 'nil
preserveDIO = 'nil
checkDIOAREA = 'nil
checkDIOPERI = 'nil
displayPININFO = 'nil
preserveALL = 'nil
```


© 2023 Cadence Design Systems, Inc.

### Description of si.env Properties

| Property | Description |
|---|---|
| `simLibName` | Name of the library containing the top-level cellview of the design. |
| `simCellName` | Name of the top-level cellname of the design. |
| `simViewName` | Name of the top-level view of the design. |
| `simSimulator` | Simulator to run. |
| `simNotIncremental` | When this property is set to `nil`, the netlister runs incrementally, which means the system netlists only the parts of your design you modified since you last netlisted the design. The default is `nil`. |
| `simReNetlistAll` | When this property is set to `t`, the netlister runs a new netlist on all the cellviews in your entire design. The default is `nil`. |
| `simViewList` | List of views to open for each cell when traversing the design hierarchy during netlisting and name translation. |
| `simStopList` | List of views that are valid stopping points for expansion used during netlisting. |
| `hnlNetlistFileName` | Name of the text netlist file. |
| `simRunDir` | Directory in which CDL data is stored. Set this global variable to the current run directory. This variable is set when the simulation environment is initialized. |
| `resistorModel` | String that sets the model name to be treated as a short. Prints out the string in the `*.RESI` command. The default is `nil`. |
| `shortRES` | Sets the value of resistance below which the resistor is assumed to be shorted. Prints the value out in the `*.RESI` command. The default is `2000.0`; type is floating point. |
| `preserveRES` | When this property is set to `t`, resistors are preserved for checking in LVS, `shortRES`, and `checkRESSIZE`. Using the optional variable [`XX`], you can specify a model name that preserves only the specified type of resistor. The default is `nil`. |
| `checkRESVAL` | Prints out `*.RESVAL` when set to `t`. The default value is `nil`. |


© 2023 Cadence Design Systems, Inc.

| Property | Description |
|---|---|
| checkRESSIZE | If preserveRES is nil, prints out *.RESSIZE when checkRESSIZE is set to t. The default is nil. |
| preserveCAP | When this property is set to t, *Export – CDL* preserves capacitors for checking in LVS. You can define checkCAPAREA if preserveCAP is t. The default is nil. |
| checkCAPVAL | Prints out *.CAPVAL when set to t. The default is nil. |
| checkCAPAREA | If checkCAPVAL is nil, prints out *.CAPAREA when checkCAPAREA is set to t. The default is nil. |
| preserveDIO | If preserveDIO is set to t, *Export – CDL* preserves the diodes for checking in LVS. You can define the variable checkDIOAREA if preserveDIO is t. The default is nil. |
| checkDIOAREA | Prints out *.DIOAREA when set to t. The default is nil. |
| checkDIOPERI | Prints out *.DIOPERI when set to t. The default is nil. |
| displayPININFO | When displayPININFO is set to t, prints out the *.PINIFO command for each subcircuit followed by the terminal names and pin directions (input, output, input/ Output). The default is nil.<br><br>If the pin information line exceeds the maximum number of characters allowed on a line, each continuation line of pin information is also preceded by *.PININFO instead of the usual line continuation character(s). |
| preserveALL | If preserveAll is set to t, resistors, capacitors, and diodes are preserved for checking in LVS. If preserveAll is set to nil, resistors, capacitors, and diodes are removed. The default is nil. |

**Note:** If you want to use the property lvsIgnore equal to FALSE on some of the instances of resistors, then you should use the SKILL variables *preserveRES* and *shortRES* as follows:

- Set the SKILL variable *preserveRES* to t by setting the toggle value of Check Resistors equal to True.
- Set the value of SKILL variable *shortRES* equal to the maximum value of all resistances below which all the resistors are to be ignored by putting the value in *Resistor Threshold Value* field.


© 2023 Cadence Design Systems, Inc.

### Defining cdlSimStopList and simStopList in the si.env File

The behavior of `cdlSimStopList` and `simStopList` properties are interdependent:

- When `cdlSimStopList` is defined with some value, the value of `simStopList` becomes the same as the value of `cdlSimStopList`.
- When only `simStopList` is defined with some value, the value of `cdlSimStopList` becomes the same as the value of `simStopList`.
- When both properties are defined, the value of `simStopList` becomes the same as the value of `cdlSimStopList`.
- When both properties are not defined, the default value `list("auCdl")` is set for both.

## Creating a config view for auCdl

To create a config view for auCdl:

1. In CIW, choose *File - New - Cellview.*

   The Create New File form appears.
2. In the *Cell Name* field, enter the name of the cell in which you want to create the config view.
3. In the *View Name* field, enter the name of the view you want to create—for example, `config_aucdl`.
4. In the *Tool* cyclic field, select *Hierarchy - Editor*.
5. Click *OK*.
6. Specify the top-level cell name and its view.
7. Click *Use Template*.

   The Use Template form appears.
8. In the *Name* cyclic field, choose *auCdl*.
9. Click *OK* to display the New Configuration form.

   You can modify the view list.
10. Click *OK* to create the view.
11. Choose *File – Save* to save the configuration.


© 2023 Cadence Design Systems, Inc.

## How to include partial netlist file in SUBCKT calls

You can automatically bind your cells to source files which will then be included in the .subckt statements.

Add following in your .simrc file

```
hnlReadHdbProps = 't
ansCdlHdbFilePathProp = "<property name>"
```

Using Hierarchy Editor, add the property to the `lib/cell/view` as cell property in which the netlist needs to be included. In the value field of this property, define full path of the partial netlist file and netlist the config view of the top cell. After the netlist is complete the information is added to the subckt file.

### Example

In the given example, you want to include the file `"/tmp/netlist/dummy_top1.net"` inside `LIB5/top1/schematic subckt`. The contents to be added are `: X17 A B / dummytop1`.

The original subckt in the netlist looks like:

```
**********************************************************************
* Library Name: LIB5
* Cell Name: top1
* View Name: schematic
**********************************************************************

.SUBCKT top1 A B
*.PININFO A:I B:O
XI0 A B / mid

.ENDS
```

In .simrc, set `ansCdlHdbFilePathProp = "abc"`

In the Hierarchy Editor for `LIB5/top1/schematic`, define a property `"abc"` with value `"/tmp/netlist/dummy_top1.net"`

After netlisting the top cell using Hierarchy Editor, the subckt file will read as follows:

```
**********************************************************************
* Library Name: LIB5
* Cell Name: top1
* View Name: schematic
**********************************************************************

.SUBCKT top1 A B
*.PININFO A:I B:O
XI0 A B / mid
```


© 2023 Cadence Design Systems, Inc.

```
.ENDS
********************************************************************************
* This auCdl Netlist has been included for cell top1:
* NOTE: The connectivity in this netlist has not been verified by auCDL
*
X17 A B / dummytop1
*********************************************************************************

.ENDS
```

### Verification

After adding the property, the Check prop.cfg should read as:

```
cell LIB5.top1
{
string prop abc = "/tmp/netlist/dummy_top1.net"
}
```

## Generating individual SUBCKT netlists

By default, auCDL netlisting generates a CDL top-level subckt netlist. This netlist comprises the netlists of all subcircuits in the design.

To generate individual subckt netlists, you need to set one of the following properties in the simrc or si.env file, based on your requirement:

- `cdlPrintIndividualSubckt = t`

  When you use this option, netlists for all individual circuits is generated in the `ihnl` directory. The files are named `cds0`, `cds1`, and so on. This feature is similar to other netlisters such as Spectre and Verilog.

  Additionally, you can set `hnlSoftLineLength` to different values, such as `72`, `80`, `90`, and so on, to specify the desired formatting of the individual subckt netlists.

  **Note:** Netlists for empty subckts and included subckts are not generated when you use this option.

- `cdlSubcktsDir = <`*`directory_name`*`>`

  When you set this property, a directory with the name you specify is created in the run directory. All individual netlists are stored in this directory with the file name as <*`cell_name`*>`.cdl`.


© 2023 Cadence Design Systems, Inc.

Additionally, the top header of each individual netlist file will include the same header as that for the top-level netlist. For example, when you use this option, all individual will comprise the following information in the header:

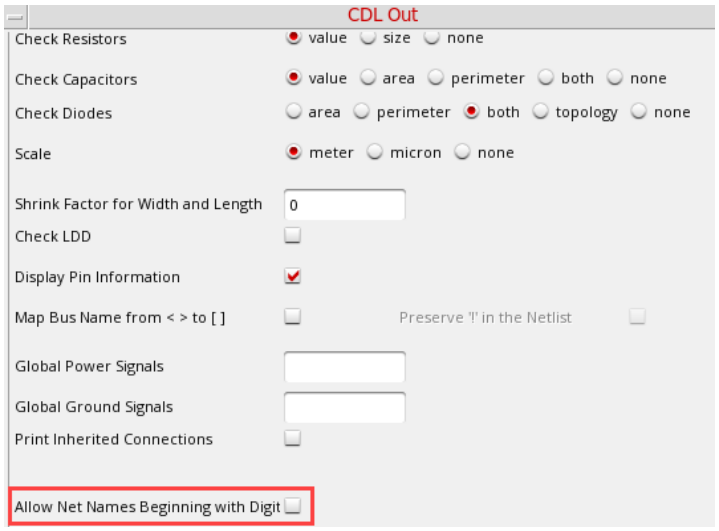
```
*.BIPOLAR
*.RESSIZE
*.CAPPERI
*.DIODE
*.EQUATION
*.SCALE METER
*.MEGA
.PARAM

*.GLOBAL soisub!

*.PIN soisub!
```

**Note:** Netlists for empty subckts is not generated when you use this option.

You can also set the `cdlPrintBusDelimiter = t` and `auCdlPParEvaluation = t` flags to further customize the generation of individual netlists.

- `allowNetNamesBeginningWithDigits = t`

  Prints net names starting with numeric characters as displayed in the schematic, without renaming them in the CDL netlist output.




  By default, the value of `allowNetNamesBeginningWithDigits` is `nil`, and net names starting with numeric characters are renamed as `N0, N1`, and so on.

  When `allowNetNamesBeginningWithDigits` is set to t, net names starting with numeric characters are printed as they appear on schematic, `1netx` and `2nety`, and not renamed as `N0, N1`.


© 2023 Cadence Design Systems, Inc.

# Customization Using the .simrc File

The behavior of the netlist can be further controlled using the simulation run control (`.simrc`) file. The parameters that you can include in the `.simrc` file are described in this section. The parameters you can set in the `.simrc` file are the same as those that are defined using the `simSetDef` SKILL function. This SKILL function defines variables only if they have not been defined previously (that is, during initialization when the `si.env` and `.simrc` files are read).

## auCdl-Specific Parameters

These auCdl parameters can be set in the `.simrc` file:

| Parameter | Description |
|---|---|
| `auCdlCDFPinCntrl = 't` | Allows CDF `termOrder` to dictate pin ordering of the top-level cell or the cell that has the auCdl view. The default is `'nil`. |
| `auCdlScale = <m>`<br>`m = "METER"` or `"MICRON"` | Prints `*.SCALE METER` or `*.SCALE MICRON`, accordingly, in netlist. When `auCdlScale` is set to `""`, it prints `*.SCALE` in netlist.<br><br>If no value is defined in the `.simrc` or `.cdsinit` environment, by default `*.SCALE METER` is printed in netlist. |
| `auCdlCheckLDD = 't` | Turns on LDD device checking by printing `*.LDD` in the netlist. The default is `'nil`. |
| `auCdlDisablePrintSubcktCDF = 't` | Disables printing of CDF parameters in `.SUBCKT` line. The default is `'nil`. |
| `auCdlHnlInstModelPropName = 'dev`*`ModelExample`* | Allows defining a property that can be used to specify the model name for instances.<br><br>**Note:** Here, *`devModelExample`* is the name of the property that can be used to specify the model name for instances. |
| `auCdlPrintNetsetForStoppingCell = t` | |


© 2023 Cadence Design Systems, Inc.

| Parameter | Description |
|---|---|
| | Enables printing of the netset for stopping cells.<br>Default value: `nil` |
| `auCdlPrintEmptySUBCKT = t` | Enables printing of empty subcircuit for all the subcircuits that are stopping cells in the netlist. Name of the empty subcircuit is either the value of CDF parameter named model, if exists, or the device cell name.<br>Default value: `nil`<br>To know more about printing empty netlist of subcircuits, refer to Printing Empty Subcircuits. |
| `auCdlPrintMultiplicityFactor = t` | |
| | Enables printing of parameter `m` irrespective of the instance parameters list for all hierarchical blocks.<br>Default value: `nil`<br>This variable does not control the printing of parameter `m` for stopping cells or text designs. |
| `auCdlDisplayPinMap = t` | Prints the PIN map statement into the subckt.<br>Default value: `nil` |
| `cdlPrintBusDelimiter = t` | |
| | Prints *.`BUSDELIMITER` in the netlist depending on pinMAP value.<br>Default value: `nil` |
| `auCdlPParEvaluation = t` | |
| | Stops the pPar evaluation in the subckt definition.<br>Default value: `nil` |
| `auCdlPParArgsOnCell = t` | |
| | Stops the pPar evaluation at the instance lines in a top cell.<br>Default value: `nil` |


© 2023 Cadence Design Systems, Inc.

| Parameter | Description |
|---|---|
| `auCdlGCPolicy = t` | Controls Garbage Collector (gc) during netlisting.<br><br>*Important*<br>This option should be used only to improve performance when netlisting a design with more than a million pins.<br><br>To further improve performance, set the `displayPININFO` property to `nil`, or deselect the *Display Pin Information* check box on the CDL Out form. |
| `auCdlPutMathExprInSingleQuotes = t` | Enables adding mathematical expressions for parameters within single quotes.<br><br>Default value: `nil` |
| `auCdlPrintAdditionalCommentsOnSubcktHeader (fp cv)` | Adds comments to the subckt header. The parameter always takes two arguments: file pointer (`fp`) and name of the cellview (`cv`).<br><br>For example:<br>`procedure(auCdlPrintAdditionalCommentsOnSubcktHeader( fp cv )`<br>`  let( (  )`<br>`  artFprintf( fp "* Following are additional comments:\n" )`<br>`  artFprintf( fp "* Create time of cv is :%s\n" cv~>createTime )`<br>`  artFprintf(fp "* fileName is %s\n" cv~>fileName )`<br>`)`<br>`  )`<br>`)` |


© 2023 Cadence Design Systems, Inc.

| Parameter | Description |
|---|---|
| `auCdlCommentStarDotConnect = t` | Adds comments to the `*.CONNECT` statements in cdl netlists.<br>Default value: `nil`<br>By default, the comments are not printed. |
| `auCdlPrintCellNameOnFooter = t` | Prints cellname at the subckt footer.<br>Default value: `nil`<br>In the following example, `.ENDS INV` is printed at footer of `SUBCKT INV` when `auCdlPrintCellNameOnFooter` is `t`.<br>`.SUBCKT INV A Z LN=NDEF GEOP=0 GEON=0 MN=1 MP=1 LP=PDEF`<br>`*.PININFO A:I Z:O`<br>`MP0 ZA VINT VINT PCH W=WP L=LP M=MP GEO=GEOP GEOMOD=GEOP`<br>`MN0 ZA VSSVBULK NCH W=WN L=LN M=MN GEO=GEON GEOMOD=GEON.ENDS`<br>`INV` |
| `auCdlNoStarInGLOBAL = t` | Prints `.GLOBAL` in the file header.<br>Default value: `nil`<br>By default, the `*.GLOBAL` is printed in the file header. |
| `auCdlNoForwardSlash = t` | Does not print forward slash (`/`) before the master name of a device.<br>Default value: `nil`<br>The following example shows when `auCdlNoForwardSlash = t` in .simrc, then forward slash (`/`) is not added before `INVD`. |


© 2023 Cadence Design Systems, Inc.

| Parameter | Description |
|---|---|
| | `.SUBCKT DELAY A Z WN=1.0 L=0.9 CNW=4.18 CPL=0.65 CPW=4.18 CNL=0.65 W=0.9 WP=1.8`<br>`XI1 A I1L I1U Z INVD WN=WN WP=WP`<br>`.ENDS DELAY`<br><br>By default `auCdlNoForwardSlash = t` and therefore forward slash (`/`) is printed before the master name of the device `INVD`.<br><br>`.SUBCKT DELAY A Z WN=1.0 L=0.9 CNW=4.18 CPL=0.65 CPW=4.18`<br>`CNL=0.65 W=0.9 WP=1.8`<br>`XI1 A I1L I1U Z  / INVD WN=WN WP=WP`<br>`.ENDS DELAY` |
| `auCdlPrintIncFileAtFooter = t` | Prints the `.INCLUDE` statements at the bottom of the netlist.<br><br>Default value: `nil`<br><br>By default, the statements are printed at the top of the netlist. |
| `auCdlReplaceAngleBracketsWithSquare = t` | Replaces the angle brackets `<>` with the square brackets `[]` while generating netlist. |
| `auCdlIncludeFileNotFoundSeverity` | Sets the severity of a message when a file specified in the `CDL_NETLIST_FILE` property is not found. Valid values are `error` and `warning`. |


© 2023 Cadence Design Systems, Inc.

| Parameter | Description |
|---|---|
| auCdlModelPrintingStyle | Prints the model name for a resistor.<br>Default value: nil<br>By default, it prints the model name with the prefix $.<br>**Examples**<br>`Rr1 O3 vssx $[rm0fm1] m=1 w=16n l=8n`<br>If auCdlModelPrintingStyle = 'Plain, the model names are printed without any prefix.<br>`Rr1 O3 vssx rm0fm1 m=1 w=16n l=8n`<br>If auCdlModelPrintingStyle = 'ModelEquals, the mode names are printed with the prefix model, as shown below:<br>`Rr1 O3 vssx model=rm0fm1 m=1 w=16n l=8n` |
| auCdlAlwaysAddPrefixInInstName | Enables simAlwaysAddPrefixInInstName variable to prefix instance names with the namePrefix of a device type based on the specified auCdl netlisting requirements.<br>**Example**<br>If you want to always add a prefix in hierarchical instance name but not in the primitive instance name then define the auCdlAlwaysAddPrefixInInstName procedure as shown below:<br>`procedure(auCdlAlwaysAddPrefixInInstName()`<br>`;;do not always add prefix to primitives`<br>`if(hnlIsAStoppingCell( hnlCurrentMaster ) then`<br>`simAlwaysAddPrefixInInstName = nil`<br>`else`<br>`;;always add prefix to hierarchical instances`<br>`simAlwaysAddPrefixInInstName = t`<br>`)`<br>`)` |
| auCdlReplaceNetWithSurvivingNet = t | |


© 2023 Cadence Design Systems, Inc.

| Parameter | Description |
|---|---|
| | Replaces the net with its surviving net as a result of shorting. |
| `auCdlTermOrderMismatchSeverity = error` | |
| | Reports an error for auCdl and stops netlisting. |
| | By default, the auCdl netlist reports a warning and continues netlisting. |

## View List, Stop List, Netlist Type, and Comments

You can use the following variables to define the standard view list, stop list, and netlist type and specify the value of the print comments flag.

| Variable | Description |
|---|---|
| `cdlSimViewList` | A list of views. The default is `'("auCdl" "schematic")` |
| `cdlSimStopList` | A list of views. The default is `'("auCdl")` |
| `cdlNetlistType` | Netlist type hierarchical (`'hnl`) or flat (`'fnl`). The default is `'hnl`. |
| `cdlPrintComments` | Print comments? Yes (`'t`) or no (`'nil`). The default is `'nil`. |

The following variables are used for instance-based switch list configuration and also can be set:

```
simInstViewListTable
simInstStopListTable
```

## Preserving Devices in the Netlist

The `si.env` file defines the following variables that determine if resistors, capacitors, diodes, or all devices must be preserved in the netlist.

```
preserveRES      preserveCAP
preserveDIO      preserveALL
```


© 2023 Cadence Design Systems, Inc.

## Removing Devices in the Netlist

During hierarchical netlisting, you can short the terminals of a device and replace that device with a surviving net. For example, you can short terminals of parasitic devices to remove them from the netlist.

The terminals of a device can be short using any of the following methods:

- Setting the `lxRemoveDeviceProperty` string property at the instance level.
- Using the `hnlUserMultiTermShortCVList` SKILL variable.
- Using the `hnlUserShortCVList` SKILL variable.

For more information on shorting devices, see Removing Devices with Two Terminals or Removing Devices with Multiple Terminals in the *Open Simulation System Reference*.

## Printing CDL Commands

The following variables let you print the associated CDL commands.

```
checkRESVAL       checkDIOAREA
checkCAPVAL       displayPININFO
checkDIOPERI      shortRES
checkRESSIZE      resistorModel
checkCAPAREA
```

## Defining Power Node and Ground Node

You can define `powerNets` and `groundNets` in the `.simrc` file. For example, if you enter the following lines in your `.simrc` file

```
powerNets  = '("VCC!")
groundNets = '("GND!" "gnd!" )
```

the auCdl netlist will show the following line:

```
*.GLOBAL VCC!:P GND!:G gnd!:G
```

**Note:** You can use the `auCdlSkipMEGA` flag for conditional printing of the `*.MEGA` statement in the auCdl netlist.This flag can be placed in the `.simrc` file, which is read by the netlister.

The `auCdlSkipMEGA` flag is used as follows:

`auCdlSkipMEGA = 'nil`
This is the default value. This enables printing of the statement in the netlist.


© 2023 Cadence Design Systems, Inc.

`auCdlSkipMEGA = 't`
When set, the `*.MEGA` statement is not printed in the auCdl netlist.

## Support for Global Power and Ground Signals from CDL UI

You can now use the Export CDL form to declare global power signal and global ground signals by following the steps given below:

1. In the CIW, choose *File – Export – CDL*.
2. In the fields, Global Power Signals field and Global Ground Signals, enter signal names respectively.

The values that you enter using the form will be added to *.GLOBAL and *.PIN statement.

:G and :P will be appended to the signal names based on the nets presence in the variables `simPowerNets` and `simGroundNets` in `.simrc` file.

## Evaluating Expressions

You might want to evaluate design variables that have been copied to the cellview using ADE and whose values are needed during verification. The Analog Expression Language mode using which auCdl evaluates expressions is determined by the setting of the SKILL environmental flag `auCdlSetTopLevelEvalMode`. Its valid values are `'t` and `nil`. The default value is `nil` and it causes auCdl to evaluate expressions by using inheritance operators. You can change the mode to full evaluation by setting the value of this flag to `'t`.

For more information on evaluation modes, refer to the Cadence document *Analog Expression Language Reference*.

## NLP Expressions

Netlisting Properties (NLP) expressions provide support for user defined properties in auCDL netlisting. You can use different NLP expressions depending on your requirements. Details about each NLP expression is described below:

- NLP expression beginning with `"[+"` is equivalent to `pPar` in AEL expression. For example, if property `"myprop"` has value `"[+subProp]"`, it will appear in auCDL netlist as `myProp = subProp`. The netlister prints the value of `subProp` for the `SUBCKT` on which it is defined.


© 2023 Cadence Design Systems, Inc.

- NLP expression beginning with "[@" is equivalent to atPar in AEL expression. For example, if property "myprop" has value "[@subProp]", it will appear in auCDL netlist as myProp = subProp.
- NLP expression beginning with "[~" is equivalent to iPar in AEL expression. For example, if property "myprop" has value "[~subProp:new value %: not found]" and subProp has a value of 10 for the instance being netlisted, it will be printed in the netlist as myProp = new value 10. However, if subProp is not defined at instance level, it will be printed in the netlist as myProp = not found.

## Mapping Global Pins

In the DFII environment, global signals in a netlist end with a ! character. If you do not want global signals to end with !, you can specify this by using either one of the following methods:

- Click *File – Export – CDL* to open the CDL Out Run form and select the *Map Pin Names from <> to []* option button.
- In the .simrc file, set the SKILL environmental variable pinMap to 't. This is a boolean variable and can have the value 't or 'nil. This variable when set to 't uses the following rules to map net names:

```
"+" -> nil
"(" -> nil
")" -> nil
"," -> nil
"/" -> nil
"." -> nil
$" -> nil
"[" -> nil
"]" -> nil
"<" -> "["
">" -> "]"
"!" -> nil
```

Default direction of global power and ground pins is INPUT.

The SKILL environmental variable hnlMapNetInName can be used similarly. For example:

```
pinMap = 't
```

is equivalent to:

```
hnlMapNetInName = list('("+" nil) '("(" nil) '(")" nil) '("," nil) '("/" nil)
'("." nil) '("$" nil) '("[" nil) '("]" nil) '("<" "[") '(">" "]") '("!" nil) )
```


© 2023 Cadence Design Systems, Inc.

To enable *Map Bus Name from <> to []* in the CDL Out form when `hnlMapNetInName` is overridden by other values in the `.simrc` file, specify the following in the `.simrc` file:

```
when(pinMAP && boundp('hnlMapNetInName)
    hnlMapNetInName = append( hnlMapNetInName list('("<" "[") '(">" "]")))
)
```

## Renaming Cell Names

You can define the `auCdlModuleNameMapFunc` SKILL variable in the `.simrc` file to rename cell names in the auCdl netlist.

For example, to add a prefix `AAA_` to the cell names in the auCdl netlist, add the following entries in the `.simrc` file:

```
auCdlModuleNameMapFunc = 'myPoCellNameMap

procedure(myPoCellNameMap( cvID )
poCellNameMap( cvID~>libName cvID~>cellName cvID~>viewName)
)

procedure( poCellNameMap(lib cell view)
   prog((mapname)
   sprintf(mapname "AAA_%s"  cell)
   return(mapname)
       )
   )
```

**Note:** Setting this variable does not detect the cell name collision and the netlister uses the same name if two cells have the same name (even if they are in different subckt).

## Renaming Pcell Subcircuits

You can define the `nlSetPcellName` SKILL procedure in the `.simrc` file to customize renaming of Pcell subcircuits in the auCdl netlist.

For more information about the `nlSetPcellName` procedure, see the *Virtuoso Analog Design Environment SKILL Language Reference*.

**Note:** When you define Pcell parameters in CDF, ensure that the type of each CDF parameter is consistent with parameter type specified in the Pcell.

## Customizing Bulk Node Search

Bulk node connection or the substrate connection on a device is specified by using the CDF property of the form (`progn bn`). If the bulk node property is found on the instance terminal,


© 2023 Cadence Design Systems, Inc.

then the net name connected to the instance terminal, which has the same name as the value of bulk node property on the instance, is printed as the bulk node connection in the netlist.

When the bulk node property is found on both instance terminal and cellview terminal, then by default, preference is given to the instance terminal. However, in this case, user can also specify how the bulk node property should be selected. For example, user can give preference to cellview terminal over instance terminal for bulk node connection by adding the following in the `.simrc` file:

```
auCdl.bulkNodeLookUp = `("cvTerm")
```

**Note:** `cvTerm` can be used only when bulknode is found on both instance terminals and cellview terminals.


© 2023 Cadence Design Systems, Inc.

# Support for HED Features

In addition to supporting the basic features of HED, auCdl also supports its following advanced features.

- Nested/Sub-Configurations – A nested configuration, also known as a sub-configuration is a configuration that is defined within another configuration. A sub-config can be nested at any level in a parent configuration.
- Occurrence Binding – Occurrence bindings are configuration rules that are defined at the occurrence level. An occurrence is an object that is defined by the full path from the top-level design to that object. In the hierarchy editor, setting any of the following attributes identifies the object as an occurrence:
  - Occurrence binding, that is, library, cell, and view binding
  - Occurrence stop point. See the subsection Occurrence Level under Stop Points.
  - Occurrence-Level Bind-to-Open. You can specify that an occurrence is unbound, that is, it is not bound to a specific library, cell or view, by setting a bind-to-open attribute on it. The bindings for the occurrence can be set later by other tools that use the configuration.
- Stop Points – A stop point on a design unit prevents the design unit from being expanded when the hierarchy is expanded. It can be applied at three levels:
  - Cell level – A stop point on a cell prevents the cell from being expanded when the hierarchy is expanded.

    **Note:** A stop point on a cell applies to all occurrences of the cell.
  - Instance (within a cell) level – You can specify a stop point on a single instance within a cell to prevent the instance from being expanded when the hierarchy is expanded. A stop point on an instance can be applied to multiple objects. If the cell that contains the instance is used in multiple places in the design, the stop point can be applied to the instance in all these places.
  - Occurrence level – An occurrence stop point is a stop point on a specific path and applies only to one instance in the design. If an object has already been defined as an occurrence, when you add a stop point you are automatically adding it to the occurrence and not to the instance.

  Cell and instance level stop points may also be specified using the `nlAction` property on a cell and instance, respectively, whereas there an occurrence stop point may be specified only through HED.


© 2023 Cadence Design Systems, Inc.

# Custom Netlisting Procedures

You can use the following netlisting procedures in the device CDF to customize how instance lines are printed within a .SUBCKT definition in the auCdl netlist.

| Netlist Procedures | Description |
|---|---|
| ansCdlSubcktCall | Prints the instance name, terminals of instance, master name, and user defined properties. |
| ansCdlCompPrim | Prints the mapped current instance names and net names. |
| ansCdlCompParamPrim | Prints the DOTMODEL property and supports all instance parameters present in *CDF - simInfo - instParams*. |
| ansCdlSpecParamPrim | Prints component name of the device if the component `param` is present in *CDF - simInfo - instParams*. |
| ansCdlSubcktCallExtended | Prints subcircuit instances and multiplier in instParams. Additionally, prints all the remaining instParameters in *CDF - simInfo.* |
| ansCdlHnlPrintInst | When specified in the CDF, the device information written in the instance lines in the auCdl netlist is customized using SKILL procedures. |

To use the netlisting procedures, do the following:

1. From CIW, choose *Tools – CDF – Edit* to open the Edit Component CDF form.
2. Select *Base* as CDF type.
3. Select master cell for the instance.
4. Select simulation information and choose auCDL as simulator.

The following section describes the format of instance lines for different netlisting procedures.

## ansCdlSubcktCall

The procedure `ansCdlSubcktCall` prints:

- instance name appended to “X”.


© 2023 Cadence Design Systems, Inc.

- terminals of instance. To change terminal order, use `auCdlCDFPinCntrl` and define termOrder in CDF simInfo section.
- cell name. In case of P cell and non-stopping cells, mapped module name is printed.
- user defined properties inherited by cells down the hierarchy.
- print instance parameters from simulation information section of the cell in `name = value` pair for stopping cell. For non-stopping cells, print ‘m’ and ‘M’ as “M=...”
- inherited connection attributes for non-stopping cells if `simPrintInhConnAttributes` is set to `t`.

**Note:** `acdlCurrentInstName()` returns mapped instance name for both hierarchical and flat netlisting.

## ansCdlCompPrim

`ansCdlCompPrim()`

### Description

The procedure `ansCdlCompPrim` is used for printing primitive devices. It prints:

- mapped current instance names with device prefix, which represents the namePrefix from auCdl simInfo of the device.
- net names on the instance in the same order of terminals as specified in device CDF termOrder. It also prints error message in case of error on CDF termOrder of the device.
- Inherited connections attributes for non-stopping cells if `simPrintInhConnAttributes` is set to `t`. If device instance has some of the inherited terminals explicitly overridden, $PINS statement is printed along with termName=netName pairs.

## ansCdlCompParamPrim

`ansCdlCompParamPrim()`

### Description

The procedure ansCdlCompParamPrim() is used for printing primitive device.

- if DOTMODEL property is present in *CDF- simINfo - instParams*, `"$.MODEL=<property_val>"` is printed.


© 2023 Cadence Design Systems, Inc.

- it supports all instance parameters present in *CDF - simInfo - instParams* even if they are not recognized.
- if some parameters are specified in *CDF - simInfo - dollarParams*, they are printed as `"$<param_value>"`.
- if some parameters are specified in *CDF - simInfo - dollarEqualsParams*, they are printed as `"$<param_name>=<param_val>"`.

**Note:** For detailed information about the usage of instparams, see chapter 4, “Modifying Simulation Information”of Component Description Format User Guide.

## ansCdlSpecParamPrim

```
ansCdlSpecParamPrim()
```

### Description

The procedure ansCdlSpecParamPrim() is used for printing primitive device. This is same as ansCdlCompPrim except the following :

- If DOTMODEL property is present in *CDF- simINfo - instParams*, `"$.MODEL=<property_val>"` is printed.
- Supports all parameters present in *CDF - simInfo - instParams* even if they are not recognized.
- If some parameters are specified in *CDF - simInfo - dollarParams*, they are printed as `"$<param_value>"`.
- If some parameters are specified in *CDF - simInfo - dollarEqualsParams*, they are printed as `"$<param_name>=<param_val>"`.
- Prints component name of the device if 'component param is present in *CDF - simInfo - instParams*.

**Note:** For detailed information about the usage of instparams, see chapter 4, “Modifying Simulation Information”of Component Description Format User Guide.

## ansCdlSubcktCallExtended

```
ansCdlSubcktCallExtended()
```


© 2023 Cadence Design Systems, Inc.

### Description

The procedure ansCdlSubcktCallExtended() is used for printing of subcircuit instances. It prints:

- mapped instance name that is prefixed with "X".
- nets on the instance in the same order as specified in *CDF - simINfo - termOrder* for device terminals.
- modelname of the device. It honors flag `auCdlPrintModelEquals`. Set `auCdlPrintModelEquals = `t` in the .simrc file, to print the string "`model=`" during explicit or implicit netlisting.
- component name of the device if 'component is present in *CDF - simInfo - instParams*.
- prints comment coded parameters, such as, $.MODEL=<model_name>, if DOTMODEL is present in CDF - simInfo - instParams. However if 'tsmcmodel is present in instParams, its value overrides in $.MODEL statement. These parameters are printed after the regular parameters. To print the comments coded parameters in the default order, set the `auCdlPrintDollarParamsInEnd` variable as `nil` in the *.simrc* file.
- prints multiplier(m-factor) if 'm or 'M is present in instParams.
- prints all remaining instParameters in *CDF - simInfo*.

**Note:** For detailed information about the usage of instparams, see chapter 4, “Modifying Simulation Information”of Component Description Format User Guide.

## ansCdlHnlPrintInst

You can specify the `ansCdlHnlPrintInst` netlist procedure in the CDF for devices and then use the following SKILL procedures and variables in the simulation run control (`.simrc`) file to customize how device information is written in the instance lines in the auCdl netlist. For more information about these SKILL procedures and variables, see the


© 2023 Cadence Design Systems, Inc.

*Virtuoso Analog Design Environment SKILL Language Reference*.

| SKILL Procedure/Variable in .simrc File | Description |
|---|---|
| `auCdlInstPrintOrder` | By default, auCdl uses the following order to write the device information for each instance in the netlist:<br>**1.** Instance name<br>**2.** Model name (for primitives only)<br>**3.** Module name (for subcircuits only)<br>**4.** Inherited parameters<br>**5.** Names of nets connected to instance terminals<br>You can customize this order using the `auCdlInstPrintOrder` SKILL variable. For more information, see auCdlInstPrintOrder on page 1265. |
| `ansCdlGetInstPrintOrder` | Reorders the printing of instance, connections, model name, instance properties, and instance parameters in netlist. |
| `ansCdlPrintInstName` | By default, auCdl prints the mapped instance name and prefixes the value of the `namePrefix` parameter in the device CDF simulation information.<br>For example, auCdl prints the instance name in the format:<br>`CC2`<br>Where `C2` is the instance name and the prefix `C` is the value of the `namePrefix` property specified in the device CDF.<br>You can customize this format using the `ansCdlPrintInstName` SKILL procedure. |


© 2023 Cadence Design Systems, Inc.

| SKILL Procedure/Variable in .simrc File | Description |
|---|---|
| `ansCdlPrintModelName` | By default, auCdl searches for model names for instances of a primitive in the following order:<br>**1.** Value of the instance property defined using `auCdlHnlInstModelPropName` parameter in the `.simrc` file. For more information about the `auCdlHnlInstModelPropName` parameter, see Customization Using the .simrc File on page 1244.<br>**2.** Value of the `'model` property on the instance.<br>**3.** Value of the `modelName` parameter in the device CDF simulation information.<br>**4.** Value of the `'componentName` property on the instance.<br>**5.** Value of the `componentName` parameter in the device CDF simulation information.<br>and prints first available model name in the following format in the netlist:<br>`model=`*`modelName`*<br>Use the `ansCdlPrintModelName` SKILL procedure to customize the order in which auCdl looks for model names for primitives and the format in which the model information is written in the netlist. |


© 2023 Cadence Design Systems, Inc.

| SKILL Procedure/Variable in .simrc File | Description |
|---|---|
| `ansCdlPrintModuleName` | By default, auCdl searches for the mapped module name for subcircuit instances in the following order:<br>**1.** The module name specified using `auCdlModuleNameMapFunc` parameter in the `.simrc` file. For more information about the `auCdlModuleNameMapFunc` parameter, see Renaming Cell Names on page 1254.<br>For Pcell subcircuits, the module name specified using the `nlSetPcellName` SKILL procedure in the `.simrc` file. For more information about the `nlSetPcellName` SKILL procedure, see Renaming Pcell Subcircuits on page 1254.<br>**2.** The module name specified for the subcircuit instance.<br>and prints the first available module name in the following format:<br>`/ modulename`<br>Use the `ansCdlPrintModuleName` SKILL procedure to customize how the module name is written in the netlist.<br>**Note:** The database objects cannot be modified by this function. |
| `ansCdlPrintInheritedParams` | By default, auCdl prints the information about inherited parameters (`pPar` properties) for a device in the following format:<br>`Property_Name=Property_Value`<br>You can customize this format using the `ansCdlPrintInheritedParams` SKILL procedure. |


© 2023 Cadence Design Systems, Inc.

| SKILL Procedure/Variable in .simrc File | Description |
|---|---|
| `ansCdlPrintInstParams` | By default, auCdl prints instance parameters in the following format:<br>*`Property_Name=Property_Value`*<br>You can customize this format using the `ansCdlPrintInstParams` SKILL procedure. |
| `ansCdlPrintInstProps` | By default, auCdl does not print user-defined instance properties in the netlist.<br>Use the `ansCdlPrintInstProps` procedure to enable printing of user-defined instance properties and also customize the format in which the properties are printed in the netlist.<br>**Note:** To print user-defined properties in the netlist, you must also use the `'instProps` argument in the auCdlInstPrintOrder SKILL variable defined in the `.simrc` file. |
| `ansCdlPrintConnections` | By default, auCdl prints the information about nets connected to a device in the following format:<br>`$PINS B=mmA_M`<br>Where `B` is the terminal name and `mmA_M` is the net name.<br>You can customize this format using the `ansCdlPrintConnections` SKILL procedure. |
| `ansCdlPrintString` | Prints comment strings in the device information for instances in the auCdl netlist.<br>**Note:** To print comment strings in the device information, you must also use the `'string` argument in the auCdlInstPrintOrder SKILL variable defined in the `.simrc` file.<br>You cannot modify the database objects using this function. |
| `auCdlPrintGlobalStmtInFooter` | |


© 2023 Cadence Design Systems, Inc.

| SKILL Procedure/Variable in .simrc File | Description |
|---|---|
| | By default, auCdl might not print all global signals in the flat netlisting mode correctly.<br><br>To print all global signals, such as `*.GLOBAL` and `*.EQUIV`, set the `auCdlPrintGlobalStmtInFooter` SKILL variable to `t` in the `.simrc` file.<br><br>**Note:** After setting this variable, `*.GLOBAL` and `*.EQUIV` appear in the netlist footer.<br><br>Default value: `nil` |
| `auCdl PrintTopCellPort` | Prints ports for the top cell in the netlist.<br><br>This option is not supported for subcircuit instances in flat auCdl netlists. |
| `auCdlFnlRetainPathInInstAndNets` | Prints the instance names with hierarchical path during the flat auCdl netlisting. |
| `fnlHierarchyDelimiter` | Prints the auCDL netlist with the specified delimiter for the node names. |

### auCdlInstPrintOrder

You can use the `auCdlInstPrintOrder` variable in the `.simrc` file to customize the order in which device information is written in the netlist.

The `auCdlInstPrintOrder` variable supports the following arguments.

| Argument | Description |
|---|---|
| *'connections* | Prints the names of nets connected to instance terminals. |
| *'inheritedParams* | Prints inherited parameters (pPar properties). |
| *'instName* | Prints the instance name. |
| *'instParams* | Prints the instance parameters (as defined in the device CDF). |


© 2023 Cadence Design Systems, Inc.

| | |
|---|---|
| *'instProps* | Prints user-defined instance properties.<br><br>By default, auCdl does not print user-defined instance properties in the netlist if you use the `'instProps` argument. You must define the `ansCdlPrintInstProps` procedure in the `.simrc` file to enable printing of user-defined instance properties in the netlist. |
| *'model* | Prints the model name (for primitives only). |
| *'moduleName* | Prints the module name (for subcircuits only). |
| *'string* | Prints comment strings in the device information.<br><br>By default, auCdl does not print comment strings in the device information. You must define the `ansCdlPrintString` SKILL procedure in the `.simrc` file to specify the comment string to be printed. For more information on the `ansCdlPrintString` SKILL procedure, see the *Virtuoso Analog Design Environment SKILL Language Reference*. |

By default, auCdl uses the following order of arguments to print the device information for an instance in the netlist.

```
('instName 'model 'moduleName 'inheritedParams 'instParams  'instProps
'connections)
```

You can customize this order by changing the order of the arguments specified for the `auCdlInstPrintOrder` variable in the `.simrc` file.

For example, you can use the `auCdlInstPrintOrder` variable to write the instance name first, followed by the names of nets connected to the instance, a comment string for the nets, the module name and a comment string for the module name, by inserting the following line in the `.simrc` file:

```
auCdlInstPrintOrder=list('instName 'connections 'string 'moduleName 'string)
```

## Netlist Customization Using netlistProcOpts

The netlistProcOpts feature allows you to specify or override any of the options provided from one of the existing functions to the internal function.

### Examples

### Example 1:


© 2023 Cadence Design Systems, Inc.

If you use `ansCdlCompParamPrim` to print `DevModel=my_model` instead of `$.MODEL=my_model`, which is a default behaviour.

Specify `netlistProcOpts = "((?printDotModel \"DevModel\"))"` in auCdl simInfo to override the default DOTMODEL syntax provided from `ansCdlCompParamPrim` to the internal printing function.

**Note:** Duplicated parentheses at the beginning and end are necessary because netlistProcOpts is a list of values, which are in the form of pairs, hence this is a list of just one pair.

### Example 2:

If you want to print $SUB after the model name in the instance line of a three-terminal resistor device, specify `(?paramOrder CDLResSUBLast)` in the *netlistProcOpts* field.The following gets printed:

```
.SUBCKT new1 n0 n1
* net n0 = /A
* net n1 = /B
* net n2 = /net1
R0 n0 n1 1e3 RP $SUB=n2
```

When the value of netlistProcOpts is not specified, the default netlist gets printed as follows:

```
.SUBCKT new1 n0 n1
* net n0 = /A
* net n1 = /B
* net n2 = /net1
R0 n0 n1 1e3 $SUB=n2 RP
.ENDS
```


© 2023 Cadence Design Systems, Inc.

The following options are available in netlistProcOpts:

| Option | Values |
|---|---|
| `(?modelStyle value)` | Changes the model style for a device.<br>■ `Omit`: Disables model printing.<br>■ `Plain`: Prints the model without any additional syntax.<br>■ `ModelEquals`: Prints the model as `model=%s`.<br>■ `DollarBrackets`: Prints the model as `$[%s]`.<br>■ `DollarDotMODEL`: Prints the model as `$.MODEL=%s`.<br>■ `DollarDotModel`: Prints the model as `$.Model=%s`. |
| `(?paramOrder value)` | Prints the value in a specific order.<br>■ `CDLResSUBLast`: Prints in the default order with SUB last for a resistor.<br>■ `instParameters`: Prints in the order specified in the CDF instParameters list. |
| `(?paramStyle value)` | Prints the parameter name and value in a specified style.<br>■ `Name`: Prints the parameter name, e.g., `R`.<br>■ `DollarName`: Prints the name prefixed by $, e.g., `$R`.<br>■ `Value`: Prints the value, e.g., `1k`.<br>■ `DollarValue`: Prints the value prefixed by $, e.g., `$1k`<br>■ `NameValue`: Prints name=value, e.g., `R=1k`.<br>■ `DollarNameValue`: Prints $name=value, e.g., `$R=1k`<br>■ `NameDollarValue`: Prints name=$value, e.g., `R=$1k`.<br>■ `DollarNameDollarValue`: Prints both prefixed by $, e.g., `$R=$1k`. |
| `(?printSUB value)` | Prints the SUB when value is non-nil. |
| `(?printNETSETandPINS value)` | Prints the $NETSET and $PINS blocks when value is non-nil. |


© 2023 Cadence Design Systems, Inc.

| Option | Values |
|---|---|
| `(?printDollarParams value)` | Prints the parameters specified in the *dollarParams* field when non-nil. |
| `(?printDollarEqualParams value)` | Prints the parameters specified in the *dollarEqualParams* field when non-nil. |
| `(?debug value)` | Prints the debug value.<br>■ `nil`: no debug logging.<br>■ `t`: Allows debug logging.<br>■ `PrintSimInfo`: Prints debugging with simulation information for each call. |


© 2023 Cadence Design Systems, Inc.

# Blackbox Netlisting

The term blackbox signifies a macro treated as a cell with only an interface definition and no internal details specified. For example, a block to be used by a customer, C, is being designed by a vendor, V. V has formally announced the characteristics of the block and passed on an interface for it to C. C should be able to netlist this block as a blackbox for initial rounds of verification and plug in the V-supplied netlist, when available, and run a final cycle of verification. This would save C time that would otherwise have been spent waiting for the block. C can specify a property on the master instance of the cell instantiated and the cell will be netlisted as a blackbox; that is, only the interface of the cell is printed in the netlist and the instances within it are skipped.

The description of the SKILL environment variable flag to enable or disable the feature is:

| | |
|---|---|
| auCdlDisableBlkBox=‘t | Disables the feature |
| auCdlDisableBlkBox=‘nil | Enables the feature |

The default value of the variable is `'nil`. This will mean that the blackbox netlisting feature is enabled, by default.

A boolean property needs to be added on the cellview that is to be treated as a blackbox. The descriptions of the valid values of this property are:

| | |
|---|---|
| auCdlPrintBlkBox=‘t | Treats the macro as a blackbox |
| auCdlPrintBlkBox=‘nil | Treats the macro as is |

The steps to be followed to work with this feature are:

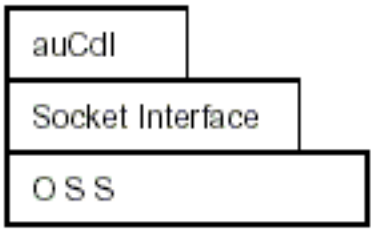
1. Ensure that in the `.simrc` file, the SKILL flag has the line `auCdlDisableBlkBox='nil`.
2. Specify the cell to be treated as a blackbox and open the Edit Cellview Properties form. Add the boolean property `auCdlPrintBlkBox` and set its value to `'t`.
3. Check and save the cellview.
4. Generate the netlist using *File – Export – CDL*.

**Note:** Set the shell variable CDS_Netlisting_Mode to Analog for auCdl netlisting.

Additionally, you can specify CDL `simInfo` to have a custom pin order for a blackbox and/or to print parameters on instance call of a blackbox.


© 2023 Cadence Design Systems, Inc.

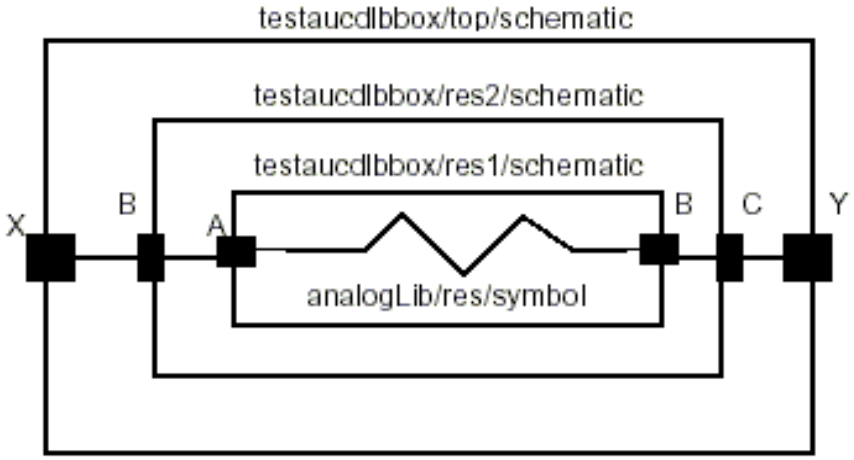
The following figure describes the location of auCdl in DFII with regard to OSS and Socket Interface.




The property to be added on the cellview is a boolean property. Any incorrect property type will be flagged as an error with the following error message in the `si.log` file:

```
Netlister Error: Incorrect property type defined for property "auCdlPrintBlkBox"
on cellview libname/cellname/viewname. The type of the property can only be a
boolean.
```

### Sample Hierarchical Cell Using Blackboxing




#### *Default Netlist*

```
*******************************************
* auCdl Netlist:
* Library Name: testaucdlbbox
* Top Cell Name: test
* View Name: schematic
* Netlisted on: Feb 6 16:32:46 2003
*******************************************
*.EQUATION
```


© 2023 Cadence Design Systems, Inc.

```
*.SCALE METER
*.MEGA
.PARAM
*****************************************
* Library Name: testaucdlbbox
* Cell Name: res1
* View Name: schematic
*****************************************
.SUBCKT res1 A B
*.PININFO A:I B:O
RR0 A B 1K $[RP]
.ENDS
*****************************************
* Library Name: testaucdlbbox
* Cell Name: res2
* View Name: schematic
*****************************************
.SUBCKT res2 B C
*.PININFO B:I C:O
XI0 B C / res1
.ENDS
*****************************************
* Library Name: testaucdlbbox
* Cell Name: test
* View Name: schematic
*****************************************
.SUBCKT test X Y
*.PININFO X:O Y:I
XI1 X Y  / res2
.ENDS
*****************************************
```

### *Netlist when auCdlPrintBlkBox=‘t on testaucdlbbox/res2/schematic:*

```
*****************************************
* auCdl Netlist:
* Library Name: testaucdlbbox
* Top Cell Name: test
* View Name: schematic
* Netlisted on: Feb 5 14:57:30 2003
*****************************************
```


© 2023 Cadence Design Systems, Inc.

```
*.EQUATION
*.SCALE METER
*.MEGA
.PARAM
*****************************************
* Library Name: testaucdlbbox
* Cell Name: res1
* View Name: schematic
*****************************************
.SUBCKT res1 A B
*.PININFO A:I B:O
RR0 A B 1K $[RP]
.ENDS
*****************************************
* Library Name: testaucdlbbox
* Cell Name: res2
* View Name: schematic
*****************************************
.SUBCKT res2 B C
*.PININFO B:I C:O
.ENDS
*****************************************
* Library Name: testaucdlbbox
* Cell Name: test
* View Name: schematic
*****************************************
.SUBCKT test X Y
*.PININFO X:O Y:I
XI1 X Y  / res2
.ENDS
*****************************************
```

Notice that the macro `res2` has been generated as a blackbox with only its interface, that is terminal information, being printed in the netlist. The difference in the netlists is marked in bold typeface.


© 2023 Cadence Design Systems, Inc.

# Additional Customizations

This section describes the following additional customizations that you can make:

- Automatically Including a Partial Netlist File within the .SUBCKT Definition for the Top or Mid-Level Cells in your Design
- Including a ROM-Insert Netlist Automatically Into the auCdl Netlist
- PININFO for Power and Ground Pins
- .PARAM Statement
- Specifying Terminal Order for auCDL Netlisting in the Edit CDF Form
- Specifying the Terminal Order for Terminals in .simrc
- Rules followed to Print Terminal Order during Netlisting
- Changing the Pin Order
- Arranging the Terminal Ordering
- Notification about Net Collision
- Making a Stop Cell at Subcircuit Level
- Printing Empty Subcircuits
- Passing Parameter
- Netlisting the Area of an npn

## Automatically Including a Partial Netlist File within the .SUBCKT Definition for the Top or Mid-Level Cells in your Design

If a block instantiated in your design has a CDL netlist, you can automatically include the CDL netlist for the block within the `.SUBCKT` definition for the top or mid-level cells in your design and disable printing of subcircuit instances in the netlist.

For example, consider a two level hierarchical design that has the following netlist. The original `.SUBCKT` definition for the top-level cell `LIB5/top/schematic` appears in the netlist as shown in bold text below:

```
*****************************************************************
* Library Name: LIB5
* Cell Name: mid
* View Name: schematic
```


© 2023 Cadence Design Systems, Inc.

```
********************************************************************************

.SUBCKT mid A B
*.PININFO A:I B:O
XI0 A B / bot
.ENDS

********************************************************************************
* Library Name: LIB5
* Cell Name: top
* View Name: schematic
********************************************************************************

.SUBCKT top A B
*.PININFO A:I B:O

XI0 A B / mid
.ENDS
```

This procedure describes how you can include a partial CDL netlist file `dummy_top1.net` located at `/tmp/netlist` within the `.SUBCKT` definition for the top-level cell `LIB5/top/schematic` and disable the printing of instances in the subcircuit in the netlist, such that netlist is created as shown below:

```
********************************************************************************
* Library Name: LIB5
* Cell Name: top
* View Name: schematic
********************************************************************************

.SUBCKT top A B
*.PININFO A:I B:O
********************************************************************************
* This auCdl Netlist has been included for cell top from file
+ '/tmp/netlist/dummy_top1.net'.
* NOTE: The connectivity in this netlist has not been verified by auCDL
*
********************************************************************************
X17 A B / dummytop1
*********************************************************************************

.ENDS
```

The bold text in the above netlist indicates the contents of the `/tmp/netlist/dummy_top1.net` file that is written within the .SUBCKT definition for the top-level cell `LIB5/top/schematic`.

**Note:** The instance and `.SUBCKT` definition for the `mid` block are not printed in the netlist.

To include the CDL netlist file `dummy_top1.net` within the `.SUBCKT` definition for the top-level cell `LIB5/top/schematic` and disable printing of the instance and `.SUBCKT` definition for the `mid` block in the netlist, do the following:


© 2023 Cadence Design Systems, Inc.

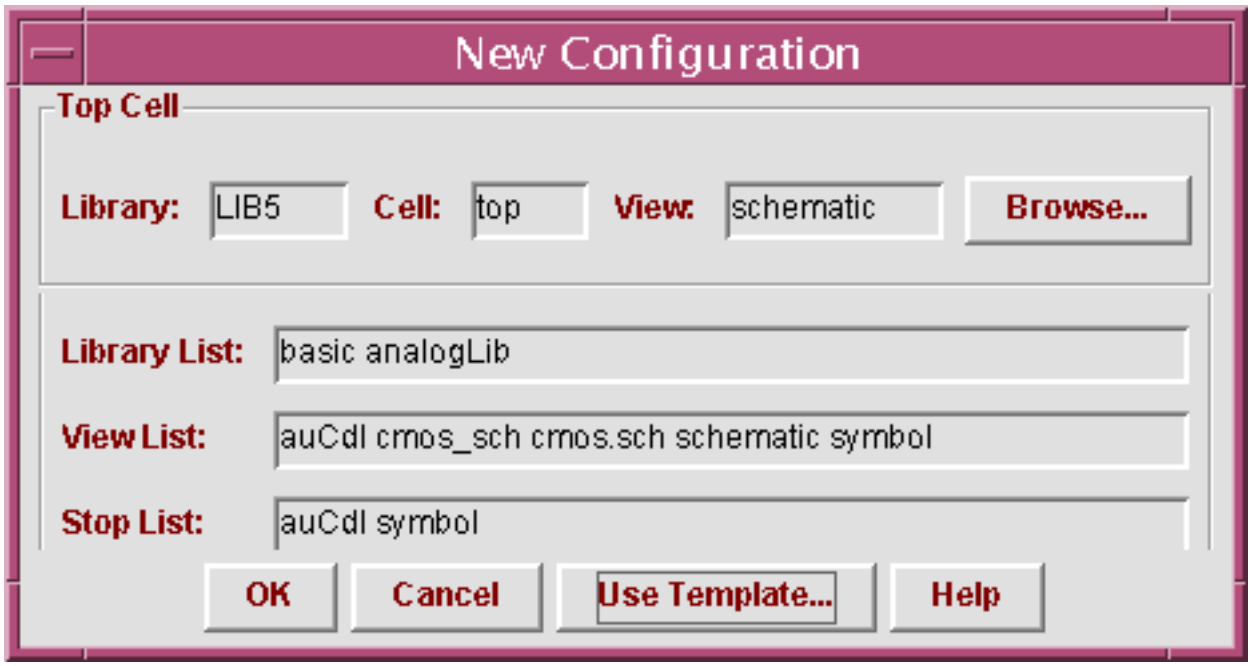
1. Add the following entries in your `.simrc` file:

   ```
   hnlReadHdbProps = 't
   ansCdlHdbFilePathProp = "<property_name>"
   ```

   **Note:** You can use any value for the `ansCdlHdbFilePathProp` variable.

2. Create an auCdl view, say, `config_aucdl`, for the top-level cell. For more information, see Creating a config view for auCdl on page 1240.

   When you create the auCdl view, ensure that the view for the top-level cell is set to `schematic`, as shown below:




   **Note:** Ensure that each instance in the design has a valid switch view. To do this, in Hierarchy Editor, ensure that the view specified for each instance in the *View to Use* column has instances in it.

3. In Hierarchy Editor, add the property specified as the value of the `ansCdlHdbFilePathProp` variable in the `.simrc` file on the schematic view of the top-level cell.

   For example, if the value of the `ansCdlHdbFilePathProp` variable is `filepath`, do the following to add the `filepath` property:

   a. In Hierarchy Editor, choose *View – Properties*.

   b. Choose *Edit – Add Property Column*.

      The Add a Property Column form appears.


© 2023 Cadence Design Systems, Inc.

**c.** Enter `filepath` in the *Property Name* field.

**d.** In the *Property Type* cyclic field, select `String`.

**e.** Click *OK* to add a column for the `filepath` property.

**f.** Right click the *filepath* column and choose Set "filepath" Cell Property.

**g.** Enter the path to the CDL netlist file for the block. For example, enter:

```
/tmp/netlist/dummy_top1.net.
```

**h.** Click *OK*.

You can verify that the property is added by opening the `prop.cfg` file in the config view for the cell. In this example, the `prop.cfg` file will contain the following text:

```
cell LIB5.top
{
string prop filepath = "/tmp/netlist/dummy_top1.net"
}
```

## Including a ROM-Insert Netlist Automatically Into the auCdl Netlist

You can include CDL netlist in top-level netlist by either concatenating the netlist or by adding `.include` statement.

- Set the SKILL environmental variable `auCdlEnableNetlistInclusion` in the `.simrc` file.
- Create an auCdl view for the instance whose netlist you want included in the top-level netlist. The view can be created by copying over the existing symbol view of the cell as the auCdl view.
- Add the cellview property `CDL_NETLIST_FILE` to the auCdl view with its `valueType` as `string` and value as either of the following:
  - the absolute path to the netlist file to be included.
  - the relative path with reference to the auCdl view of the corresponding cell.
- Set the value of `auCdlEnableNetlistInclusion` to one of the values:
  - `auCdlEnableNetlistInclusion='t or "embedAtBottom"`

    Appends the netlist content in `CDL_NETLIST_FILE` to the bottom of top level netlist.


© 2023 Cadence Design Systems, Inc.

- `auCdlEnableNetlistInclusion="includeAtTop" or "includeAtBottom"`

  Adds `.include` statement in top-level netlist at top or bottom respectively.

- The default value of `auCdlEnableNetlistInclusion` is `nil`, which means the file specified by `CDL_NETLIST_FILE` is not considered.

***Related Topics***

Edit Object Properties – Instance and Block

## PININFO for Power and Ground Pins

If you want power and ground pin names to appear with `:P` and `:G`, respectively, in the *.PININFO line in the CDLOut netlist for non-global signals, you can specify this with the `cellViewPowerPins` and `cellViewGroundPins` properties.

For example, you may have four pins in the cellView, namely `A`, `B`, `VSS`, and `VDD`, and you want the PININFO lines to appear as follows:

```
.SUBCKT test A B VDD VSS
*.PININFO B:P VSS:G A:G VDD:P
.ENDS
```

From the schematic cellView, click *Edit – Properties – CellView*. Click *Add* in the *User Property* section and add the following properties:

- `cellViewPowerPins`, with *Type* as `ilList` and *Value* as `("B" "VDD")`
- `cellViewGroundPins`, with *Type* as `ilList` and *Value* as `("A" "VSS")`

Then, check and save the cellView.

When you run the netlister, CDL Out checks for two properties of the type `ilList` in the cellview, namely `cellViewPowerPins` and `cellViewGroundPins`, and generates the netlist according to information specified with them. The PININFO lines in the netlist appear as mentioned above.

## .PARAM Statement

The design variables specified on Top cell of the design being netlisted will be printed in `.PARAM` statement each per line.


© 2023 Cadence Design Systems, Inc.

For example, if the `designVarList` property specified on top cell has the following value:

```
( ("CAP" "0.8p") ("RES" "20") ("X" "35") )
```

the `.PARAM` will be printed as:

```
.PARAM CAP=0.8p
+ RES=20
+ X=35
```

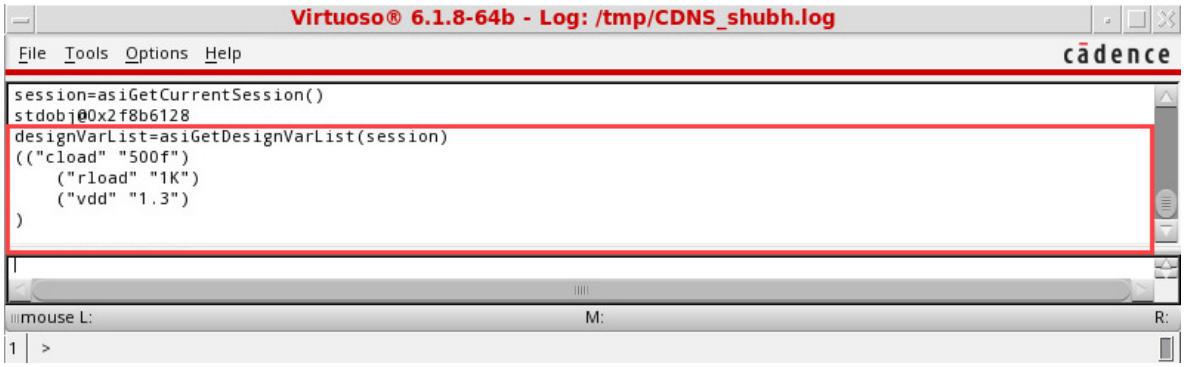
To use the `designVarList` property, follow these steps:

1. Open a maestro view with one or more design variables.
2. Get the session id as follows:

   ```
   session=asiGetCurrentSession()
   ```

3. Use the following call to get the list of design variables:

   ```
   designVarList=asiGetDesignVarList(session)
   ```




## Specifying Terminal Order for auCDL Netlisting in the Edit CDF Form

To specify the terminal order:

1. In CIW, click *Tools – CDF – Edit*.
2. Specify the library and cell names.
3. Set *CDF Type* to *Base.*
4. Open the *Simulation Information* tab.
5. In the *Choose Listing* field, select *By Simulator* and then select *auCdl* in the drop-down list.


© 2023 Cadence Design Systems, Inc.

6. In the *termOrder* field, enter the terminals in the order in which you want them in the netlist.
7. Click *Apply* and *OK* to close the implement the changes and close the dialog box.

For hierarchical netlisting:

1. Set the SKILL flag `auCdlCDFPinCntrl` to '`t` in the `.simrc` file. For more information, see The .simrc File.
2. Build the netlist using auCdl.

## Specifying the Terminal Order for Terminals in .simrc

You can define the `auCdlCDFPinCntrl` parameter in the `.simrc` file to have custom terminal order for a cell.This parameter supports various combinations using the values `cdfTermOrder`, `masterTermOrder`, and `userDefinedFunctionNameSymbol`. The following table shows how you can define the `auCdlCDFPinCntrl` parameter:

`auCdlCDFPinCntrl = nil`

Prints terminals in alphanumerical order. This is the default order.

`auCdlCDFPinCntrl = t / auCdlCDFPinCntrl = "cdfTermOrder"`

Reads the CDF terminal order, if available, else uses the default order.

`auCdlCDFPinCntrl = "masterPortOrder"`

Uses the port order of the master cell, if available, else uses the default order.

```
auCdlCDFPinCntrl = (list "cdfTermOrder" "masterPortOrder"
'userDefinedFunctionNameSymbol)
```


© 2023 Cadence Design Systems, Inc.

Reads the terminal order in the following sequence based on availability:

1. CDF terminal order for the device
2. Master port order
3. User-defined function to get the terminal order for the device
4. Default terminal order, if the values mentioned above are unavailable

**Note:** You can change the order of elements or drop elements in the list.

```
auCdlCDFPinCntrl = (list "cdfTermOrder" "masterPortOrder"
"warnAndDefault")
```

Reads the terminal order in the following sequence based on availability:

1. CDF terminal order for the device
2. Master port order, if the CDF terminal order is unavailable
3. Default order and displays the auCDL-56 warning message, if the values mentioned above are unavailable

```
auCdlCDFPinCntrl = (list "cdfTermOrder" "masterPortOrder" "error")
```

Reads the terminal order in the following sequence based on availability:

1. CDF terminal order for the device
2. Master port order, if the CDF terminal order is unavailable

If both CDF terminal order and master port order are not available, displays the auCDL-50 error message and stops netlist generation.

```
auCdlCDFPinCntrl = '("cdfTermOrder" "masterPortOrder"
"default"|"alphanum")
```

Disables the auCDL-50 message when both CDF terminal order and master port order are not available, and `default` or `alphanum` are specified in the list.


© 2023 Cadence Design Systems, Inc.

## Rules followed to Print Terminal Order during Netlisting

When `auCdlCDFPinCntrl` is set to `t` or `"cdfTermOrder"`, the following rules are obeyed during netlisting, when:

**Terminal order in CDF is missing**: The default terminal list is used to print the netlist for that cell or instance.

**Terminal order is CDF has fewer terminals than the default terminal list**: The terminals are printed in the following sequence:

1. The terminals specified according to the order specified in the CDF terminal order.
2. The remaining terminals sorted alphanumerically.
3. The inherited terminals.

**Note:** For stopping cells (leaf-level cells), the remaining terminals are not printed.

For example, consider there are ten terminals, and only six terminals have been specified in CDF termOrder. In this case, during netlisting these six terminals are printed according to the CDF terminal order and the remaining four terminals are printed alphanumerically. Finally, any inherited terminals are printed in the netlist.

**Terminal order has duplicate terminals**: A warning message is issued as described in the section Error Handling and the terminals are printed in the following way:

- For non-leaf level cells, the terminal order is ignored and the default terminal list is used for netlisting.
- For stopping cells (leaf level cells), the terminals are printed in the terminal order and are followed by the inherited terminals, if any.

**Terminal in the terminal order is not valid**: A warning message is issued as described in the section Error Handling and the default terminal list is used for netlisting.

**Note:** Similar rules are followed when when `auCdlCDFPinCntrl` is set to `t` or `"masterPortOrder"`, where instead of CDF terminal order, master port order is followed to print the terminals. You can also specify any of the following additional existing options to control the terminal order of bus members:

- Individual members of a bus to be specified in any order in the termOrder
- Split buses to be specified in any order in the termOrder

**Note:** If termOrder is empty, the default terminal list is used.

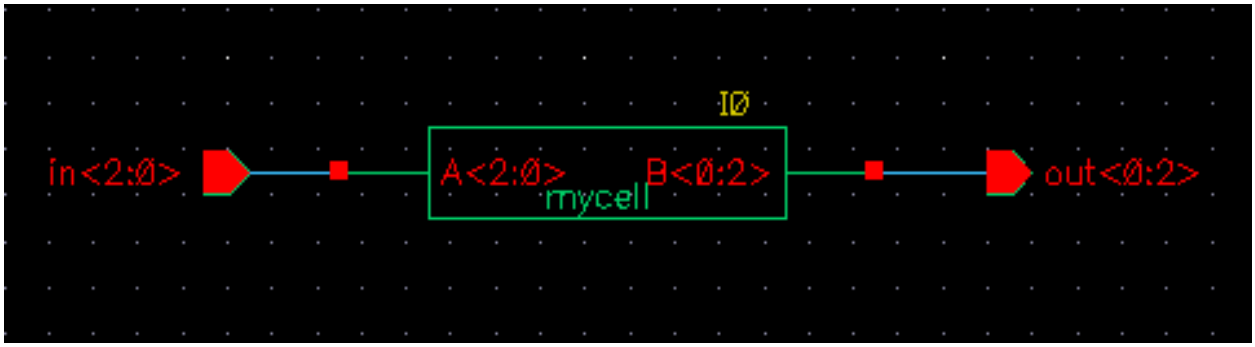
**Note:** Set the shell variable `CDS_Netlisting_Mode` to Analog for auCdl netlisting.


© 2023 Cadence Design Systems, Inc.

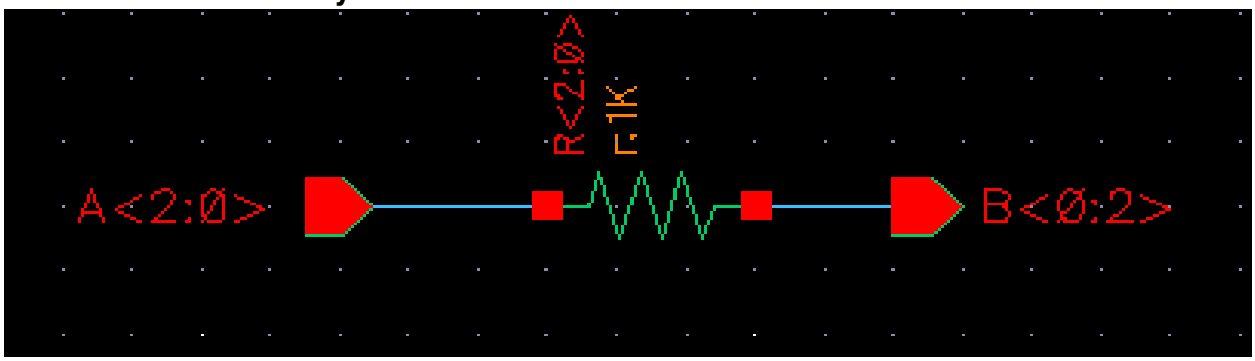
### Example for CDF termOrder

Consider a hierarchical design of the cell `mytop` using `mycell` as a sub-cell. Here, `mycell` has been set as a stopping cell to make the example compact.

### Schematic View for mytop




### Schematic View for mycell




Assuming that the top schematic is `mytop`, consider the following cases:

### Default netlist (No termOrder Is Specified)

```
********************************************************************************
* auCdl Netlist:
*
* Library Name: mylib
* Top Cell Name: mytop
* View Name: schematic
* Netlisted on: Apr 10 14:31:28 2003
********************************************************************************
*.EQUATION
*.SCALE METER
*.MEGA
.PARAM
```


© 2023 Cadence Design Systems, Inc.

```
********************************************************************************
* Library Name: mylib
* Cell Name: mytop
* View Name: schematic
********************************************************************************
.SUBCKT mytop in<2> in<1> in<0> out<0> out<1> out<2>
*.PININFO in<2>:I in<1>:I in<0>:I out<0>:O out<1>:O out<2>:O
XI0 in<2> in<1> in<0> out<0> out<1> out<2> / mycell
.ENDS
```

### Using the CDF termOrder Features

For case 1, termOrder is specified as follows:

- For mytop: "in<0:1>" "out<2:1>
- For mycell: "A<1:0>" "B<0:1>"

### Case 1: Missing Terminals in termOrder

```
********************************************************************************
* Library Name: mylib
* Cell Name: mytop
* View Name: schematic
********************************************************************************
.SUBCKT mytop in<0> in<1> out<2> out<1> in<2> out<0>
*.PININFO in<2>:I in<1>:I in<0>:I out<0>:O out<1>:O out<2>:O
XI0 in<1> in<0> out<0> out<1> / mycell
.ENDS
```

Two points to note here are:

- As mytop is not a leaf-level cell, the terminals in the termOrder are followed by the missing terminals in the netlist.
- As mycell is a leaf-level cell, the missing terminals will not be printed in the netlist.

### Case 2: Invalid Terminal

When `auCdlCDFPinCntrl` is set to ’t and termOrder for `mytop` is set as:

```
"in<0:1>"  "out<2:1>" "T"
```

TermOrder will be ignored and the default terminal list will be printed for `mytop` along with the warning message.

```
********************************************************************************
* Library Name: mylib
```


© 2023 Cadence Design Systems, Inc.

```
* Cell Name: mytop
* View Name: schematic
********************************************************************************

.SUBCKT mytop in<2> in<1> in<0> out<0> out<1> out<2>
*.PININFO in<2>:I in<1>:I in<0>:I out<0>:O out<1>:O out<2>:O
XI0 in<1> in<0> out<0> out<1> / mycell
.ENDS
```

`si.log` has the following warning message:

```
*Warning* Could not determine the node name for terminal '"T"'. This may be
caused by an error in the CDF specified on:
                component   : mytop
                in cellview : schematic
                of library  : mylib
```

### Case 3: Duplicate Terminal

When `auCdlCDFPinCntrl='t` and termOrders are set as follows:

- For mytop: "in<0:1>" "out<2:1>" “in<0>”
- For mycell: "A<1:0>" “B<1>” "B<0:1>"

Note the use of individual bus bit “in<0>” and “B<1>” in the termOrder for `mytop` and `mycell`, respectively. When the termOrder is expanded, they become duplicate terminals.

```
********************************************************************************
* Library Name: mylib
* Cell Name: mytop
* View Name: schematic
********************************************************************************
.SUBCKT mytop in<2> in<1> in<0> out<0> out<1> out<2>
*.PININFO in<2>:I in<1>:I in<0>:I out<0>:O out<1>:O out<2>:O
XI0 in<1> in<0> out<1> out<0> out<1> / mycell
.ENDS
```

The `si.log` file has the following warning message

```
*Warning* Could not determine the node name for terminal '"in<0>"'. This may
be caused by an error in the CDF specified on:
                component   : mytop
                in cellview : schematic
                of library  : mylib
```


© 2023 Cadence Design Systems, Inc.

```
*Warning* Could not determine the node name for terminal '"B<1>"'. This may be
caused by an error in the CDF specified on:
                component   : mycell
                in cellview : schematic
                of library  : mylib
```

The points to be noticed here are:

- As `mytop` is not a leaf-level cell, the termOrder is ignored and the default terminal list for `mytop` is printed in the netlist.
- As `mycell` is a leaf-level cell, duplicate terminals are allowed in the termOrder.

### Error Handling

A warning message is generated if there are invalid or duplicate terminals in the terminal order. The message informs that the node name for the terminal could not be determined because of an error in the CDF. The message also lists the information regarding the component (cell name), cellview (view name), and library (library name) in which the error occurred.

## Changing the Pin Order

You need to do the following to modify the pin order:

1. In the `SimInfo` section of CDF for the auCdl view, add the following lines to the file.

   ```
   netlistProcedure:   ansCdlSubcktCall
   componentname:      subcircuit
   termOrder:          "my_pin_1" "my_pin_2" "my_pin_3"
   namePrefix:          X
   ```

2. Add the following line to the `.simrc` file:

   ```
   auCdlCDFPinCntrl = t
   ```

   If a `.simrc` file does not exist, you need to create one, add the above line, and save the file in your current directory. For more information, see The .simrc File.

**Note:** If the `auCdlCDFPinCntrl = 't` variable is specified in the `.simrc` file and you want the explicit inherited terminals of stop cells to be printed in the netlist, you must specify the explicit inherited terminal names in the CDF `termOrder` field. If you want explicit inherited terminals of stop cells to be printed in the netlist even when they are not specified in the CDF `termOrder` field, add the following line in the `.simrc` file:

```
auCdlNoInhTermInTermOrder = 't
```


© 2023 Cadence Design Systems, Inc.

## Arranging the Terminal Ordering

You can use the `auCdlTermOrderStr` variable to arrange the netlist in ascending or descending order when the `auCdlCDFPinCntrl` parameter is set to `t`.

The possible values for the `auCdlTermOrderStr` variable that you can define in the `.simrc` file or specify in the CIW are:

`auCdlCDFPinCntrl=t` and `auCdlTermOrderStr = "a"/"A"/'A/'a`, then the `SUBCKT` line is printed in ascending order.

- `auCdlCDFPinCntrl=t` and `auCdlTermOrderStr = "d"/"D"/'D/'d`, then the `SUBCKT` line is printed in descending order.

**Note:** By default, `auCdlTermOrderStr` remains unbound and works only when `auCdlCDFPinCntrl` is set to non-nil.

**Note:** `auCdlTermOrderStr` displays an error in the following cases:

- When values other than `"A"`, `'A`, `"a"`, `'a`, `"D"`, `'D`, `"d"` and `'d` are specified for the `auCdlTermOrderStr` variable.
- When `auCdlCDFPinCntrl` is set to `nil`.

## Notification about Net Collision

Sometimes a net name may get mapped to a new name, such as when there are invalid characters in the original name. This new name may collide with another existing or mapped net name. Due to this collision, one of the net names is mapped to a new name.

To ensure that you get warnings or error messages for such collisions and mapping, set the SKILL variable `simCheckNetCollisionAction` as per the following table:

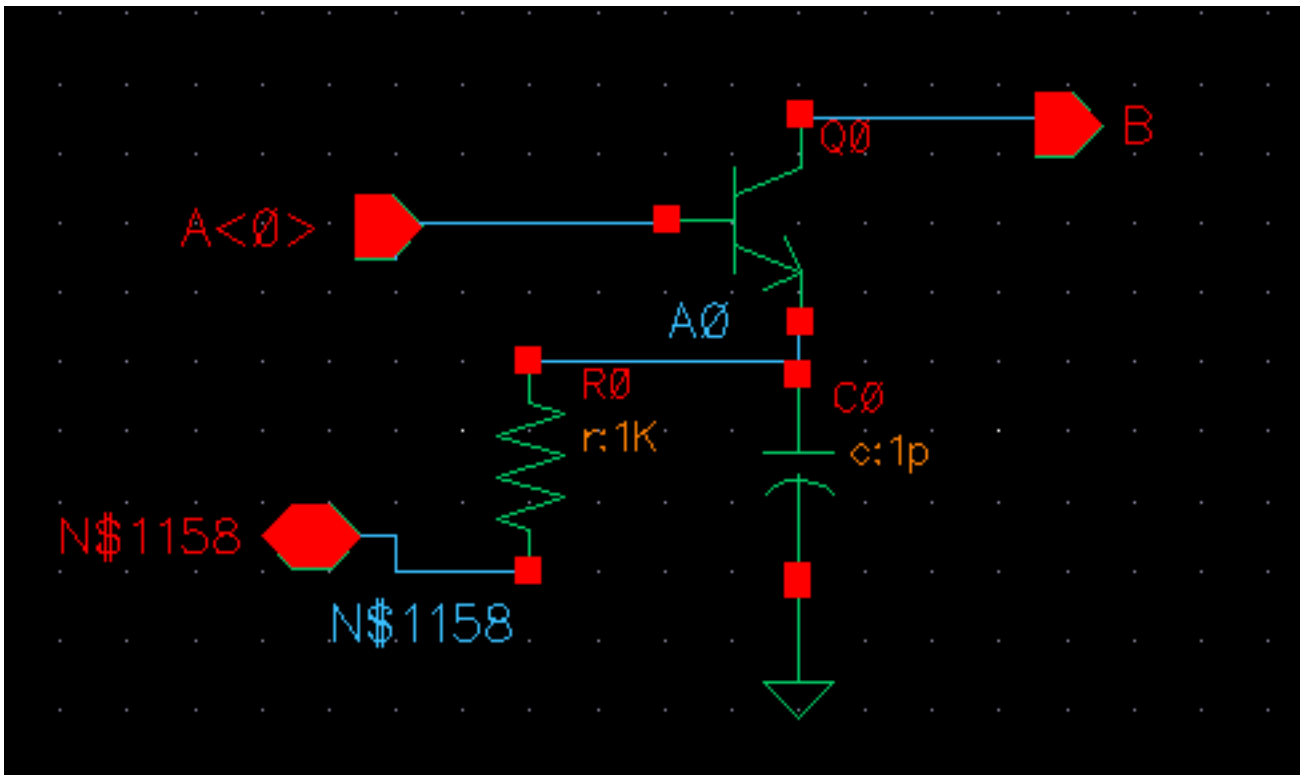
| Value | Actions Taken by Netlister |
|---|---|
| `warning` | **1.** Generates a warning message for each net name collision:<br>`WARNING: Netlister : Net <netname> has collided with an existing net name and will be remapped to <new name>`<br>**2.** Remaps collided nets.<br>**3.** Creates the netlist. |


© 2023 Cadence Design Systems, Inc.

| Value | Actions Taken by Netlister |
|---|---|
| `error` | **1.** Generates an error message in case of net name collision:<br>`ERROR: Netlister : Net <netname> has collided with an existing net name. Exiting...`<br>**2.** Stops the netlist. |
| Any value other than `warning` or `error` | **1.** Does not generate any warning or error message.<br>**2.** Remaps collided nets.<br>**3.** Creates the netlist. |

If you want the `simCheckNetCollisionAction` to operate in the batch mode or the background mode, set it in the `.simrc` file. If you want it to operate in the foreground mode, set it in the CIW.

Consider the following schematic view of the `autest` cell of a hierarchical design `mycdltest`.




Assume that the .simrc file is set as follows:


© 2023 Cadence Design Systems, Inc.

```
hnlMapNetInName = '( ("<" "") (">" ""))
simNetNamePrefix = "M"
```

The auCdl netlist obtained is as shown below.

**Note:** `A<0>` is mapped to `A0` because the `hnlMapNetInName` variable set in `.simrc`. So, it collides with the original net `A0`. After collision, the original net is mapped to `M0` because `simNetNamePrefix` is set to `M`.

```
********************************************************************************
* auCdl Netlist:
*
* Library Name: mycdltest
* Top Cell Name: autest
* View Name: schematic
* Netlisted on: Apr 21 16:12:26 2003
********************************************************************************
*.EQUATION
*.SCALE METER
*.MEGA
.PARAM
*.GLOBAL gnd!
*.PIN gnd!
********************************************************************************
* Library Name: mycdltest
* Cell Name: autest
* View Name: schematic
********************************************************************************
.SUBCKT autest A0 B N$1158
*.PININFO A0:I B:O N$1158:B
RR0 M0 N$1158 1K $[RP]
CC0 M0 gnd! 1p $[CP]
QQ0 B A0 M0 NP
.ENDS
```

### Case 1

When `simCheckNetCollisionAction` is set to `warning` and the file `.simrc` has the following settings:

```
hnlMapNetInName = '( ("<" "") (">" ""))
simNetNamePrefix = "M"
simCheckNetCollisionAction="warning"
```


© 2023 Cadence Design Systems, Inc.

the netlist generated is the same as mentioned earlier but the log file has the following message:

```
Running Artist Hierarchical Netlisting ...
WARNING: Netlister : Net 'A0' has collided with an existing net name, will be
remapped to M0.
End netlisting <Date Time>
```

### Case 2

When `simCheckNetCollisionAction` is set to `error` and the file `.simrc` has the following settings:

```
hnlMapNetInName = '( ("<" "") (">" ""))
simNetNamePrefix = "M"
simCheckNetCollisionAction="error"
```

a netlist is not generated and the log file has the following message:

```
Running Artist Hierarchical Netlisting ...
ERROR: Netlister : Net 'A0' has collided with an existing net name, exiting...
End netlisting <Date Time >
"Netlister: There were errors, no netlist was produced."
```

## Making a Stop Cell at Subcircuit Level

To make the netlister stop at the subcircuit level for a specific block (and to prevent it from netlisting down to the primitive cells for the given block), copy the symbol view of the `subckt` to an auCdl view. Then make the following modification to the `.simrc` file:

```
cdlsimViewList = list( "auCdl" "symbol" "schematic" )
cdlsimStopList = list( "auCdl" )
```

By default, the netlister does not print subcircuits for stop cells. To know how to print empty subcircuits, refer to Printing Empty Subcircuits.

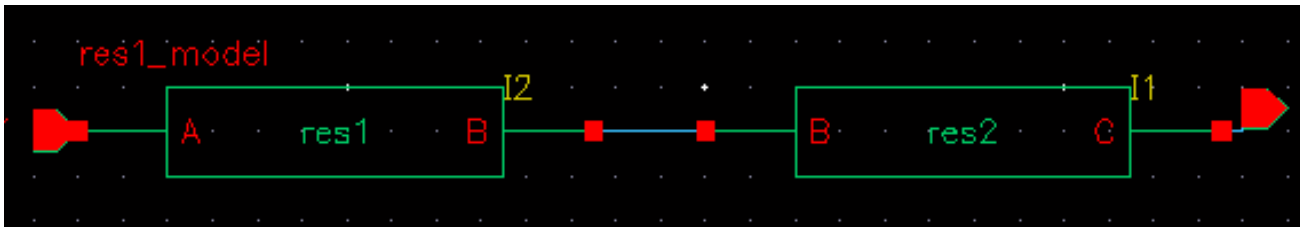
## Printing Empty Subcircuits

This section describes the procedure to print empty subcircuits for various cases.


© 2023 Cadence Design Systems, Inc.

### Printing empty subcircuits for Stopping Cells

By default, the netlister does not print subcircuits for stop cells, but you can set the `auCdlPrintEmptySUBCKT` variable to print an empty subcircuit. Consider the following schematic view of the cell, `test`.




In this example, cell `res1` is a stop cell. if you set the `auCdlPrintEmptySUBCKT` variable as `t` in the *.simrc* file, the generated netlist appears as follows:

```
.SUBCKT res2 B C
*.PININFO B:I C:O
.ENDS


.SUBCKT test X Y
*.PININFO Y:I X:O
XI1 net4 X / res2
XI2 Y net4 res1_model
.ENDS


.SUBCKT res1_model A B
.ENDS
```

In this netlist, an empty subcircuit is printed for cell `res1`. Here, `res1_model` is the model name applied to the cell.

*Important*

If a model name is applied to a cell and it is overridden at the instance level, the netlist shows the overridden name at the instance line. However, the subcircuit definition shows only the default model name, which is applied at the cell level. Therefore, it is recommended not to change the model name at the instance level.

If the stop cell has CDF, and has non-empty termOrder defined in auCdl simInfo section of CDF, terminals in termOrder are printed as ports when writing empty subcircuit definition.

However, when a stop cell or primitive has no CDF available, or does not have auCdl simInfo section in device CDF, auCdl prints device (cellview) terminals as ports when writing empty subcircuit definition.


© 2023 Cadence Design Systems, Inc.

Even when CDF/simInfo is available for device, but termOrder field in simInfo section is empty, cellview terminals are written in empty subcircuit definition.

When termOrder is not available, and cellview terminals are written, they are sorted such that first non-inherited terminals are alphanumerically sorted and printed, followed by inherited terminals alphanumerically sorted and printed.

Also, any vector/bus terminals in cellview are expanded and printed.

**Note:** If you are using a primitive netlist procedure for an empty cell, setting `auCdlPrintEmptySUBCKT` to `t` generates an auCDL-55 warning message. To avoid this warning message, use `ansCdlSubcktCall` or a non-primitive netlist procedure for empty cells.

### Printing Empty Subcircuits for Blank Cells

To print the filler cell instances, which have blank or empty cells, add the following environment variable in the appropriate `.simrc` file:

```
hnlEmptySwitchMasterAction="honor"
```

As a result, these cells get printed in the netlist:

```
.SUBCKT mycell IN OUT
*.PININFO IN:I OUT:O
MM1 net1 net3 net4 net2 PM
MM0 net5 net7 net8 net6 PM
X10 / filler
.ENDS
```

Considering that the instance switch view does not have any pins and instances, no subckt gets printed for this cell in the netlist. To get an empty subcircuit in the netlist, add the following environment variable in the appropriate .simrc file:

```
auCdlPrintEmptySUBCKT = 't
```

As a result, empty subcircuit gets printed in the netlist:

```
.SUBCKT mycell IN OUT
*.PININFO IN:I OUT:O
MM1 net1 net3 net4 net2 PM
MM0 net5 net7 net8 net6 PM
X10 / filler
```


© 2023 Cadence Design Systems, Inc.

```
.ENDS

.SUBCKT filler
*.PININFO
.ENDS
```

## Passing Parameter

Parameters can be passed to daughter cells of a subcircuit by passing `m` (M factor) to the MOS transistors that make up an inverter.

on the parent inverter: `m = 2`
on the MOS transistors:

MOS: `m = pPar("m")`
PMOS: `m = pPar("m")`

In the evaluation of a parameter, if the value of another parameter is to be incorporated, then it can be done by using the following method:

If the parameter `AD` of a MOS transistor is to be a function of its channel width, `AD` can be defined as

```
AD = iPar("w")*5u
```

## Netlisting the Area of an npn

To add a CDF parameter called Emiter Area (`EA`) to the CDF of your `npn`, fill out the CDF form with the following values:

```
paramType = string
parseAsNumber = yes
units = don't use
parseAsCEL = yes
storeDefault = no
name = EA
prompt = EA
defValue = iPar("area")
...
```

If you do not want to display the parameter on the form, you can set `display = nil`.


© 2023 Cadence Design Systems, Inc.

# CDF Simulation Information for auCdl

The auCdl netlisting procedure `ansCdlCompPrim` supports the following devices: `FET`, `CAP`, `IND`, `DIODE`, `BJT`, `RES`, and `MOS`. To use CDL Out to generate the correct name for the component, its terminal, and parameters, you need to attach auCdl CDF simulation information (`siminfo`) to cells. This can be set using *Tools – CDF – Edit* menu commands and then choosing the library/cell.

The `dollarParams` and `dollarEqualParams` fields specify the parameters whose values have to be printed with a dollar ($) prefix.

The parameters specified in the `dollarParams` section are used to print the values of these parameters with a $ sign prefixed with the value. For example, if the `dollarParams` field contains `param1`, whose value on the instance `L0` of type `inductor` (or its master or the library) is `value1`, then the netlist contains the instance statement as given below

```
LL0 net1 net2 $value1
```

The parameters specified in the `dollarEqualParams` are used to print the values on the corresponding instance, its master, or its library along with parameters with the $ prefix. For example, if the `dollarEqualParams` field in the CDF simInfo section contains `param1`, whose value on the instance `L0` of type `inductor` or on its master or the library is `value1`, then the statement for the instance in the netlist is as follows:

```
LL0 net1 net2 $param1=value1
```

The values for the `dollarParams` and `dollarEqualParams` fields use the following precedence: the Instance value overrides the Master value, which overrides the Library value.

To print `modelName` with a $ sign prefixed to it, add the parameter `TSMCMODEL` in the `instParameters` dialog box in the auCdl – simInfo section. The same precedence as specified for the `dollarParams` and `dollarEqualParams` fields is used for the model value. For example, if the instance value of `TSMCMODEL` has a value `LP` of the type `String`, then the corresponding instance line in the netlist will contain the model description as:

```
LL0 net1 net2 $.MODEL=LP
```

The following is a comprehensive list of auCdl `siminfo` for all the supported devices.

## Device CDF Values

| **FET** | |
|---|---|
| `netlistProcedure` | `ansCdlCompPrim` |


© 2023 Cadence Design Systems, Inc.

| FET | |
|---|---|
| instParameters | W L model |
| componentName | fet |
| termOrder | D G S |
| propMapping | nil W w L l m |
| namePrefix | j |
| modelName | NJ |
| dollarParams | param1, param2, param3 |
| dollarEqualParams | param1, param2, param3 |

| CAP | |
|---|---|
| netlistProcedure | ansCdlCompPrim |
| instParameters | C L W area SUB m |
| componentName | cap |
| termOrder | PLUS MINUS |
| propMapping | nil C c L l W w area a |
| namePrefix | C |
| modelName | CP |
| dollarParams | param1, param2, param3 |
| dollarEqualParams | param1, param2, param3 |

**Note:** If you specify any or all of the following: C, area and L & W , the netlister will output only one of them by using the following sequence of priority: C, area, L & W.

| IND | |
|---|---|
| netlistProcedure | ansCdlCompPrim |
| instParameters | L tcl tc2 nt ic |
| componentName | ind |
| termOrder | PLUS MINUS |


© 2023 Cadence Design Systems, Inc.

| IND | |
|---|---|
| propMapping | nil L l |
| namePrefix | L |
| modelName | LP |
| dollarParams | param1, param2, param3 |
| dollarEqualParams | param1, param2, param3 |

| DIODE | |
|---|---|
| netlistProcedure | ansCdlCompPrim |
| instParameters | area SUB pj m) |
| componentName | diode |
| termOrder | PLUS MINUS) |
| propMapping | nil |
| namePrefix | D |
| modelName | DP |
| dollarParams | param1, param2, param3 |
| dollarEqualParams | param1, param2, param3 |

| BJT | |
|---|---|
| netlistProcedure | ansCdlCompPrim |
| instParameters | W L SUB M EA m |
| componentName | bjt |
| termOrder | C B E |
| propMapping | nil EA area |
| namePrefix | Q |
| modelName | NP |
| dollarParams | param1, param2, param3 |
| dollarEqualParams | param1, param2, param3 |


© 2023 Cadence Design Systems, Inc.

| RES | |
|---|---|
| netlistProcedure | ansCdlCompPrim |
| instParameters | R SUB W L m |
| componentName | npolyres |
| termOrder | P1 P2 |
| propMapping | nil SUB sub R r W w L l |
| namePrefix | R |
| modelName | RP |

| Subcircuits | |
|---|---|
| netlistProcedure | ansCdlSubcktCall |
| componentName | subcircuit |
| termOrder | in out |
| propMapping | nil L l |
| namePrefix | X |
| dollarParams | param1, param2, param3 |
| dollarEqualParams | param1, param2, param3 |

| MOS | |
|---|---|
| netlistProcedure | ansCdlCompPrim |
| instParameters | m L W model LDD NONSWAP |
| componentName | mos |
| termOrder | D G S progn(bn) |
| propMapping | nil L l W w |
| namePrefix | M |
| modelName | |
| dollarParams | param1, param2, param3 |
| dollarEqualParams | param1, param2, param3 |


© 2023 Cadence Design Systems, Inc.

## Netlist Examples

Here are some netlist examples:

| Type | Example |
|---|---|
| Two Terminal CAP | `CC5 n3 gnd! 1p $[CP] M=10` |
| Three Terminal CAP | `CC32 n5 gnd! 1p $[CP] $SUB=n5 M=3` |
| Two Terminal RES | `RR8 n2 gnd! 1.2K $[res.mod] $W=4 $L=10 M=3` |
| Two Terminal IND | `LL1 n1 n3 1000 $[LP] $SUB=gnd!` |
| Three Terminal RES (4.3.4) | `RR3 n1 n4 1000 $[RP] $W=20 $L=100 $SUB=gnd! M=3` |
| Three Terminal RES (4.4.x) | `RR3 n1 n4 1000 $SUB=gnd! $[RP] $W=20 $L=100 M=3` |
| Diode | `DD6 a gnd! DP 10 3 M=12` |
| FET | `JJ7 d g gnd! fet.mod W=3 L=2 M=2` |
| BJT | `QQ4 c b gnd! NP M=12 $EA=100 $W=4 $L=3` |
| MOS | `MM1 g d gnd! gnd! nmos.mod W=3 L=2 M=2` |

**Note:** `auCdl` has been enhanced such that while printing the instance of a cell whose switch master is a stopping view, the `instParameters` specified in the CDF siminfo section are also printed.

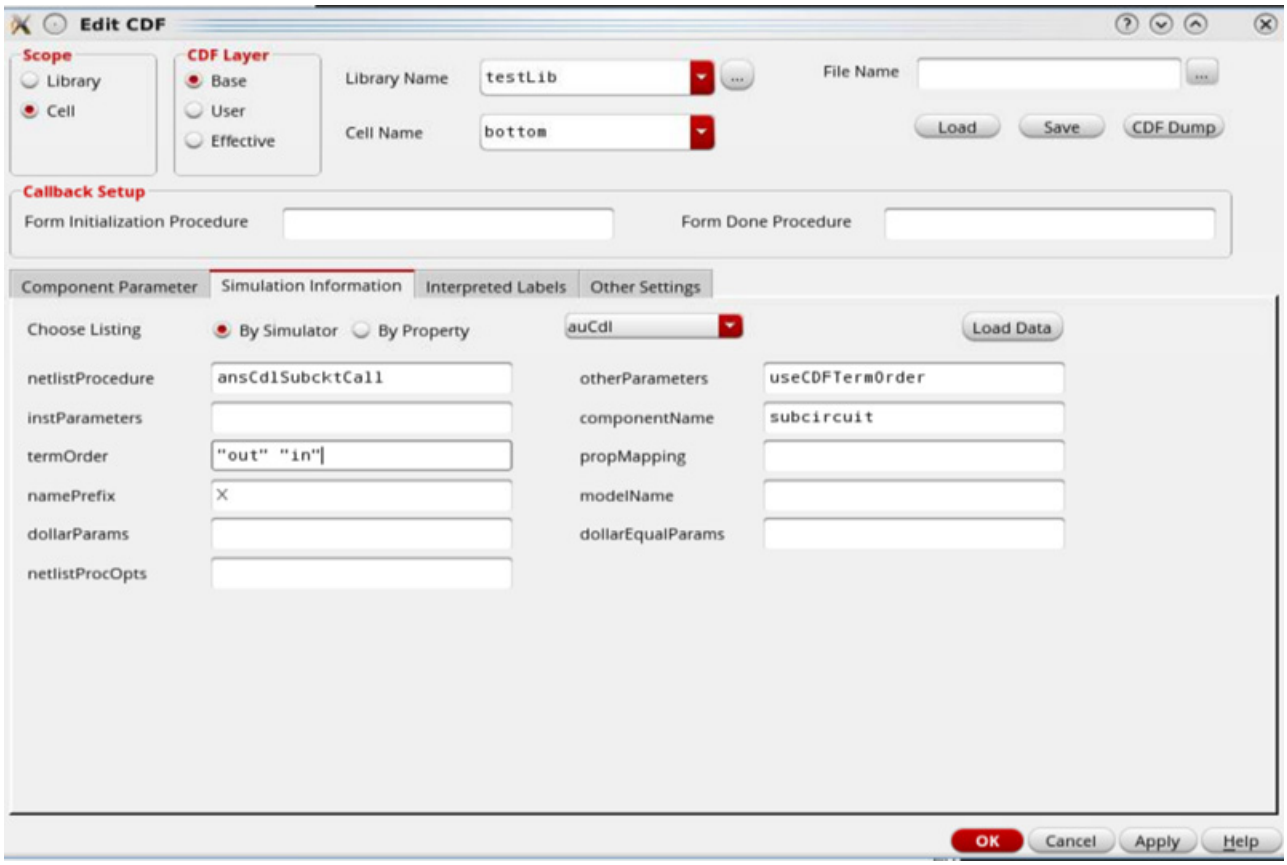
### Support of Inherited Connection on Device Substrate

In such situations, the extra terminal (the third terminal on devices like resistors, capacitors etc. or the fourth terminal on devices like transistors) is found on the stopping view rather than the symbol view (instantiated view). So the substrate connection is resolved by finding the net attached to the first extra terminal on the stopping view in comparison to `termOrder` in the CDF.

**Note:** In case of devices of type MOS, if `progn(bn)` is in the `termOrder`, then precedence would be given to `progn(bn)` and SUB would not be printed at all. Therefore for MOS devices, in order to use inherited connections on a substrate, you have to remove `progn(bn)` from the `termOrder` of the `siminfo` section of the base CDF of the device.


© 2023 Cadence Design Systems, Inc.

**Note:** If you specify `useCDFTermOrder` in the *otherParameters* field of the auCdl `siminfo` of a cell, the connectivity order is according to the `CDFtermOrder` for this cell and its instance calls.




## What is Different in the 4.3 Release

An auCdl netlist can be extracted by following these steps:

1. In the CIW, click *File – Export – CDL*
2. In the CDL Out Run form, fill in the appropriate fields and click *OK* or *Apply*.

For more information about using CDL Out, read the *Translating CDL Files* section in the *Design Data Translator's Reference*.


© 2023 Cadence Design Systems, Inc.

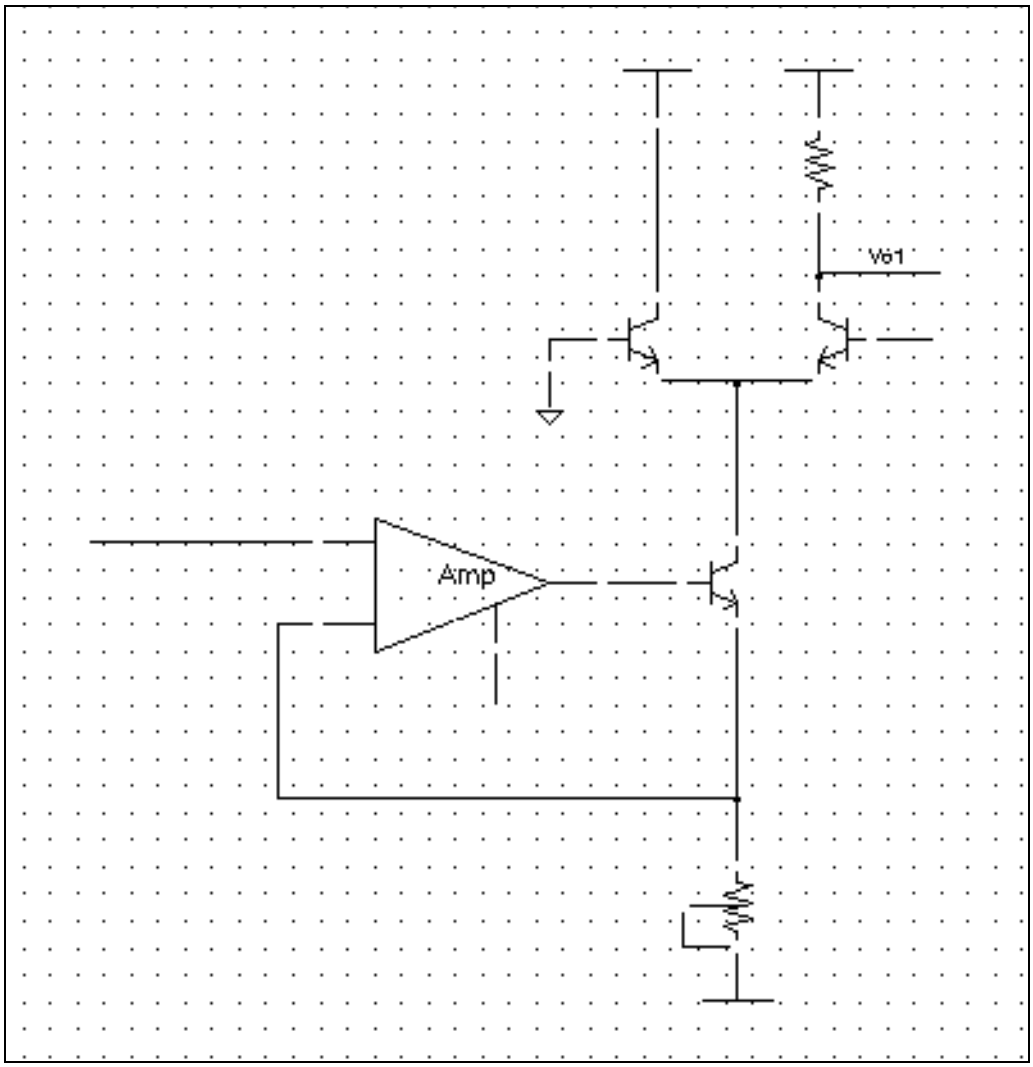
The following `si.env` parameters are used in the 4.3.x release only.

| Parameter | Description |
|---|---|
| `simLibConfigName` | The name of the configuration that determines the versions of cellview used in the design hierarchy. The default is `nil`. |
| `simVersionName` | Name of the top-level version of the design. The default value is `nil`. |
| `simLibPath` | Specifies the library search path for the library that contains both the top-level cellview and the global cellview. |


© 2023 Cadence Design Systems, Inc.

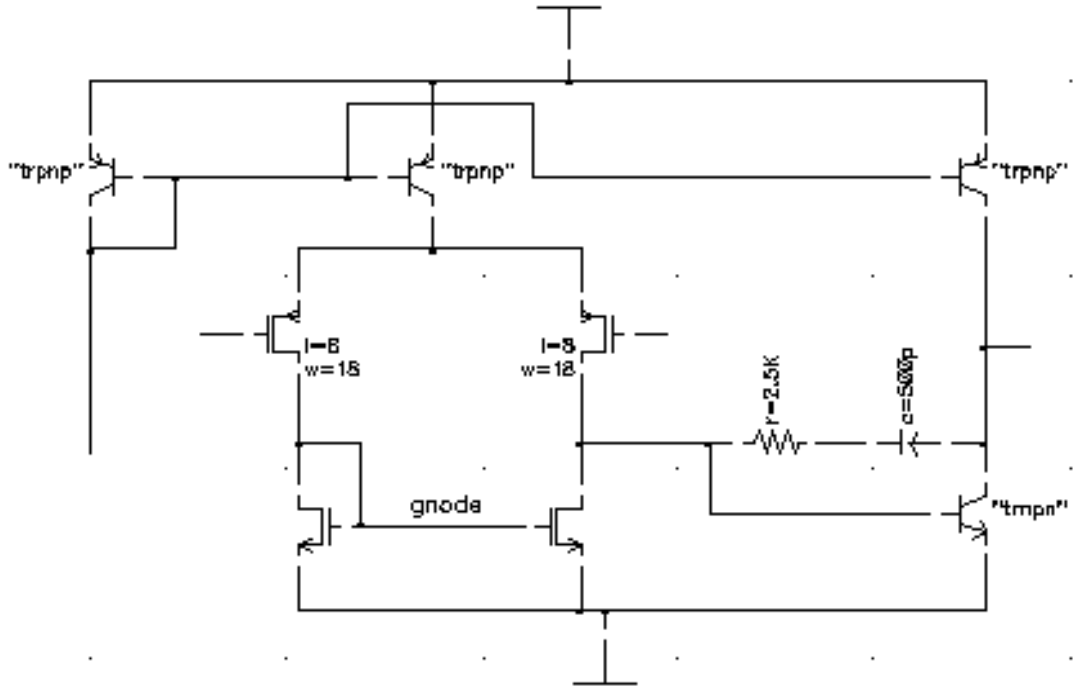
# Complete Example

The following example shows the schematic captures and the auCdl netlists.





© 2023 Cadence Design Systems, Inc.




This is the auCdl netlist.

```
*****************************************************************
*auCdl Netlist:
*
* Library Name: test_auCdl
* Top Cell Name: AMckt.auCdlonly
* View Name: schematic
* Netlisted on: Nov 1 16:12:40 1997
*****************************************************************
*.BIPOLAR
*.RESI = 2000 resmod
*.RESVAL
*.CAPVAL
*.DIOPERI
*.DIOAREA
*.EQUATION
*.SCALE METER
.PARAM

*.GLOBAL vdd!
+ vss!
+ vcc!
+ vee!
+ gnd!

*.PIN vdd!
*+ vss!
*+ vcc!
```


© 2023 Cadence Design Systems, Inc.

```
*+ vee!
*+ gnd!

************************************************************
* Library Name: test_auCdl
* Cell Name: amplifier
* View Name: schematic
************************************************************
.SUBCKT amplifier inm inp iref out
*.PININFO inm:I inp:I iref:I out:O
RR0 net52 net6 2.5K $[RP]
CC0 net6 out CAP $[CP]
QQ0 out net52 vss! NP M=1
MM1 net52 inp net26 vdd! PM W=128e-6 L=8u M=1
MM3 gnode inm net26 vdd! PM W=128u L=8e-6 M=1
MM5 gnode gnode vss! vss! NM W=100u L=10u M=1
MM2 net52 gnode vss! vss! NM W=100u L=10u M=1
QQ4 out iref vdd! PN
QQ2 iref iref vdd! PN
QQ3 net26 iref vdd! PN
.ENDS

************************************************************
* Library Name: test_auCdl
* Cell Name: AMckt.auCdlonly
* View Name: schematic
************************************************************

.SUBCKT AMckt.auCdlonly Iref Vlo Vo1 Vs
*.PININFO Iref:I Vlo:I Vo1:O Vs:I
XI3 net28 Vlo Iref net9 / amplifier
QQ2 net15 net9 net28 NP M=1.0
QQ1 vcc! gnd! net15 NP M=1.0
QQ0 Vo1 Vs net15 NP M=1.0
RR0 vcc! Vo1 10e3 $[RP]
RR1 net28 vee! 4e3 $SUB=vee! $[RP]
.ENDS
```


© 2023 Cadence Design Systems, Inc.

© 2023 Cadence Design Systems, Inc.

# B

# Environment Variables

This appendix describes public environment variables that control the characteristics of the Virtuoso ADE Explorer. You can customize the operation and behavior of ADE Explorer by changing the value of a particular environment variable.

This appendix lists the environment variables in the following areas:

- Simulation Environment
- LBS Distributed Processing Environment
- adexl.monte
- AMS
- ams.emirOpts
- ams.envOpts
- ams.netlisterOpts
- asimenv
- asimenv.monte
- asimenv.netlist
- auCore.misc
- auCore.selection
- HspiceD
- license
- maestro.cpupdtr
- maestro.distribute
- maestro.emir
- maestro.explorer


© 2023 Cadence Design Systems, Inc.

- maestro.gui
- maestro.license
- maestro.mpt
- maestro.plotting
- maestro.polo
- maestro.simulation
- maestro.stimuli
- maestro.stimuliTool
- maestro.variables
- maestro.window
- Spectre
- Spectre.emirOpts
- Spectre.envlpOpts
- Spectre.envOpts
- Spectre.hb
- Spectre.opts
- Spectre.pss
- Spectre.tran
- Spectre X
- spectreVerilog
- Ultrasim
- voltus_fi.results
- Real Time Tuning Environment Variables
- Environment Variables


© 2023 Cadence Design Systems, Inc.

# Simulation Environment

- allowAdePanicStateSaving
- allowInvalidObjectSelection
- artistPlottingMode
- autoPlot
- checkInstanceBindings
- createCDFdata
- createComponentName
- createNamePrefix
- createNetlistProc
- createOceanScriptBeforeSimulation
- defaultHierSave
- directPlotPlottingMode
- displayPointNeltlistTimeStamp
- doNotOverwriteVarValuesWhileCopyingFromCV
- drlBufferMemory
- findVariablesSkip
- ignoreSchModified
- immediatePlot
- immediatePrint
- keepShellVarsInModelLibPath
- useHierarchicalSelection
- inductorDevicesNames
- obsoleteWarnings
- oceanScriptFile
- outputsImportExportVersion


© 2023 Cadence Design Systems, Inc.

- postSaveOceanScript
- preferTermOrderOverPinOrder
- preSaveOceanScript
- projectDir
- retainStateSettings
- saveDefaultsToOCEAN
- shortInstanceList
- showConvertNotifyDialog
- simulationDate
- simulator
- strictCDFparseAsNumber
- tryNFSSync
- tryNFSSyncLocal
- useDisplayDrf
- variables
- ignorePrintNettype
- skipModelSectionsRetrieval


© 2023 Cadence Design Systems, Inc.

# LBS Distributed Processing Environment

These sections describe the environment variables used to set up and use distributed processing for job simulation in Virtuoso® using the Cadence© Load Balancing Software (LBS).

- Simulator-Specific Environment Variables
- Environment Variables for Distributed Processing
- Environment Variables for Distributed Processing with LBS

## Simulator-Specific Environment Variables

Environment variables that you add in your `.cdsinit` file are applicable to all the simulators that you want to use. To specify environment variables that apply only to specific simulators, define the `sessInitTrigFunc` function in your `.cdsinit` file.

**Note:** Ensure that the name of this function does not begin with a lower-case character.

After the `sessInitTrigFunc` function is defined in the `.cdsinit` file, you can create an entry for this function in the `.cdsenv` or `.cdsinit` files to indicate the specified function name.

Consider the following example in which you specify different resource requirement strings for the Spectre simulator and UltraSim simulator.

Add the following in your `.cdsinit` file:

```
procedure(CCSsessInitTrigFunc(session)
    case(asiGetSimName(session)
        ("spectre"
    envSetVal("asimenv.distributed" "lsfResourceString" 'string
"lsfStringForSpectre")
        )
        ("UltraSim"
    envSetVal("asimenv.distributed" "lsfResourceString" 'string
"lsfStringForUltraSim")
        )
    )
)
```

Here,

- `session` is the name of the `asi` session.
- `lsfStringForSpectre` is the LSF resource requirement string for the Spectre simulator.


© 2023 Cadence Design Systems, Inc.

- `lsfStringForUltraSim` is the LSF resource requirement string for the UltraSim simulator.

Set the following environment variables in your `.cdsenv` and `.cdsinit` files, indicating the name of the `sessInitTrigFunc` function to be used:

In the `.cdsenv` file:

```
asimenv.startup sessInitTrigFunc string "CCSsessInitTrigFunc"
```

In the `.cdsinit` file:

```
envSetVal("asimenv.startup" "sessInitTrigFunc" 'string "CCSsessInitTrigFunc")
```

## Environment Variables for Distributed Processing

This section describes the environment variables that control various distributed processing settings. Add the variables you want to use in your `.cdsenv` file.

The following environment variables are common to all distributed jobs:

- userDefinedJobName
- copyModeDir
- hostMode
- autoJobSubmit
- bypassNetlist
- hostName
- startTime
- mailTo
- stateFile
- showMessages
- startDay
- expTime
- expDay
- timeLimit
- emailNotify


© 2023 Cadence Design Systems, Inc.

- logsInEmail
- daysBeforeExpire
- block
- loginShell
- puttogetherqueue
- copyNetlist
- logsDir
- mailAllLogs
- logsInPsf
- shellCmd
- listMode
- deleteJob
- removeJobData

### userDefinedJobName

The default value is `nil`.By default, job are started with the prefix "job" in the name. When specified before a `virtuoso` run, this variable overrides the default job value in the *Job Name* field of the Job Policy Setup form. This will be effective until you explicitly change the job name in the Job Policy Setup form. You cannot set this variable during a session. This variable is most useful in regression runs with distributed systems and with different job names in each run set through the environment variable.

To set this variable in the `.cdsenv` file, enter the following:

```
asimenv.distributed userDefinedJobName string "" nil
```

#### *Example*

```
envSetval "asimenv.distributed" "userDefinedJobName" 'string "robert"
```

### copyModeDir

The directory, relative to the execution host, that is set up as the working directory of a copy mode job.


© 2023 Cadence Design Systems, Inc.

To set this variable in the `.cdsenv` file, enter the following:

```
asimenv.distributed copyModeDir string "/tmp" nil
```

### hostMode

Makes the distributed processing option the default in the analog circuit design environment.

To set this variable in the `.cdsenv` file, enter the following:

```
asimenv.startup hostMode "distributed"
```

### autoJobSubmit

If set to `nil`, the Job Policy Setup form is presented to the user at the time when the job is submitted. If set to `t`, jobs are submitted automatically.

To set this variable in the `.cdsenv` file, enter the following:

```
asimenv.distributed autoJobSubmit boolean t
```

### bypassNetlist

If this variable is set to its default value, `nil`, netlisting is done for all the subtasks of a multi-task job before job submission. If it is set to `t`, netlisting is bypassed for subtasks.

To set this variable in the `.cdsenv` file, enter the following:

```
asimenv.distributed bypassNetlist boolean nil
```

### hostName

Sets the default host name. If unspecified, the load balancer automatically selects the host.

To set this variable in the `.cdsenv` file, enter the following:

```
asimenv.distributed hostName  string    nil
```

**Note:** If the environment variables `queueName`, `hostName`, and `drmsCommand` are specified in the `.cdsinit` file, then the mode is selected as per the order of precedence – queue, list, and command. This means that if all the environment variables are specified, the queue mode will be selected; if `hostName` and `drmsCommand` are specified, the list mode will be used; whereas, if only the `drmsCommand` is specified, the command mode will be used.


© 2023 Cadence Design Systems, Inc.

### startTime

Sets the specified time as the default start time for a job. If unspecified, the job runs immediately.

To set this variable in the `.cdsenv` file, enter the following:

```
asimenv.distributed startTime  string  "20:00"  nil
```

### mailTo

Sets the default list of users who will receive an e-mail when the job terminates. If unspecified, no e-mail notification is provided.

To set this variable in the `.cdsenv` file, enter the following:

```
asimenv.distributed mailTo  string  ""  nil
```

### stateFile

By default, the system creates a file named `.adpState` in your home directory containing the job server's state. You can rename the file and put it in a different directory by modifying the statefile string.

To set this variable in the `.cdsenv` file, enter the following:

```
asimenv.distributed stateFile  string  "statefile"  nil
```

### showMessages

If set to `t`, a message is displayed in the CIW or OCEAN terminal when a job is completed.

To set this variable in the `.cdsenv` file, enter the following:

```
asimenv.distributed showMessages boolean  t  nil
```

### startDay

Sets the default start day for a job. If set to today, the job will always run on the same day it is submitted.

To set this variable in the `.cdsenv` file, enter the following:

```
asimenv.distributed startDay  cyclic  "day"  nil
```


© 2023 Cadence Design Systems, Inc.

Valid Values: `today, Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, and Saturday.`

### expTime

Sets the specified time as the default expiration time for the job. If unspecified, expiration time is based on the value of the `timeLimit` variable.

To set this variable in the `.cdsenv` file, enter the following:

```
asimenv.distributed expTime string  "06:30"  nil
```

### expDay

Sets the default expiration day for a job. If set to today, the job always runs on the same day it is submitted.

To set this variable in the `.cdsenv` file, enter the following:

```
asimenv.distributed expDay cyclic  day  nil
```

Valid Values: `today, Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, and Saturday.`

### timeLimit

Sets the default time limit for a job. If set to none, no time limit is imposed. If set to unspecified, the expiration time is based on the value of expTime and expDay variables.

To set this variable in the `.cdsenv` file, enter the following:

```
asimenv.distributed timeLimit cyclic  time  nil
```

Valid Values: `unspecified, none, 5 minutes, 15 minutes, 30 minutes, 1 hour, 3 hours, 6 hours, 12 hours, 1 day, 2 days, 3 days, 5 days, and 10 days.`

### emailNotify

If set to `t`, e-mail notification is provided following job termination.

To set this variable in the `.cdsenv` file, enter the following:

```
asimenv.distributed emailNotify boolean  t  nil
```


© 2023 Cadence Design Systems, Inc.

### logsInEmail

If set to `t`, stdout and stderr logs are included in the termination e-mail.

To set this variable in the `.cdsenv` file, enter the following:

```
asimenv.distributed logsInEmail boolean  t  nil
```

### daysBeforeExpire

Specifies the number of days after which terminated jobs are deleted from the job server.

To set this variable in the `.cdsenv` file, enter the following:

```
asimenv.distributed daysBeforeExpire int  3  nil
```

### block

If set to `t`, analog circuit design environment or OCEAN will be blocked until all the jobs have completed.

To set this variable in the `.cdsenv` file, enter the following:

```
asimenv.distributed block boolean  nil
```

### loginShell

Specifies login shell for the job. If set to none, the user's local environment is copied over to the execution host and used as the job's environment.

To set this variable in the `.cdsenv` file, enter the following:

```
asimenv.distributed loginShell  cyclic   shell  nil
```

Valid Values: `none, csh, ksh, and sh.`

### puttogetherqueue

Used to specify the queue to be used for Put Together Job.

To set this variable in the `.cdsenv` file, enter the following:

```
asimenv.distributed puttogetherqueue string "" n
```


© 2023 Cadence Design Systems, Inc.

### copyNetlist

Used to specify whether the netlist directory needs to be copied from the execution host to the submission host. This may be required during simulation if some files are generated under netlist dir.

To set this variable in the `.cdsenv` file, enter the following:

```
asimenv.distributed copyNetlist boolean nil
```

### logsDir

Used to specify the directory name where log files of jobs submitted through distributed processing are stored. The default value of the variable is "/tmp". There are some specific checks made for the value specified for `logsDir`, which are as follows:

- If the value specified starts from "." or "~", it is expanded using `simplifyFilename`().
- If a directory named `logsDir` exists, it is used as the log directory for the current job.
- Otherwise, display a message requesting for creation of a directory. If the user response is a No, the job is not submitted. If the user response is a Yes, create a directory as specified in `logsDir`:
  - If the directory is created successfully, use this as logsDir.
  - Else, display a message stating that the directory cannot be created. In this message ask if the user wishes to use "/tmp" as logsDir and then continue. If the response is a `Yes`, "/tmp" used as logsDir. If the response is a `No`, it stops at this point without any job submission.The value of this variable can be changed from one job to another.

**Note:** The user must ensure that the logsDir specified is exported on all the hosts participating in distributed processing.

To set this variable in the `.cdsenv` file, enter the following:

```
asimenv.distributed logsDir string "/tmp" nil
```

### mailAllLogs

If set to `t`, sends out a mail after completion of all the tasks and each individual task.

To set this variable in the `.cdsenv` file, enter the following:

```
asimenv.distributed mailAllLogs boolean  nil
```


© 2023 Cadence Design Systems, Inc.

### logsInPsf

If set to `t`, stores the stdout/stderr log file in the corresponding psf directory.

To set this variable in the `.cdsenv` file, enter the following:

```
asimenv.distributed logsInPsf boolean  nil
```

### shellCmd

Specifies the default value for Shell_Cmd_at_Finish as available in the Job Policy Setup form.

To set this variable in the `.cdsenv` file, enter the following:

```
asimenv.distributed shellCmd string ""
```

### listMode

Specifies whether the list option for specifying hosts needs to be displayed in the Job Policy Setup form. If set to `nil`, the list option will not be displayed. Therefore, you will only be able to submit jobs from queues.

To set this variable in the `.cdsenv` file, enter the following:

```
asimenv.distributed listMode boolean t
```

### deleteJob

Specifies whether to remove a job and its log files from the distributed processing database after the job is complete. If this variable is set to `t`, the same job name can be used in successive runs. For example, if you submit a job by the name *`Job001`*, the data for this job is generated in the `<>/..../job001` directory. When you submit another job after the completion of *`job001`*, the same job name can be used in the *Job Name* field of the Job Policy Setup form and the data for that is generated at the same directory `<>/..../job001`.

**Note:** The results of the previous run are not deleted. However, results of all the runs for the same job name are saved in the same directory. To delete the data specifically, use the *Results – Delete* menu option.

The GUI option corresponding to the `deleteJob` environment variable is the *Reuse Job Name After Completion* option on the Job Policy Setup form. To set this variable in the `.cdsenv` file use the following call:

```
asimenv.distributed deleteJob boolean t
```

To set this variable in the `.cdsinit` file or CIW, use the following call:


© 2023 Cadence Design Systems, Inc.

```
envSetVal("asimenv.distributed" "deleteJob" 'boolean t)
```

Remote login is required in order to remove the error or log files. You can delete the error or log files without the remote login, by setting the following variable in `.cdsenv` file:

```
asimenv.distributed deleteWithoutRemLogin  boolean  t
```

If set to `t`, the log files are removed from the NFS mounted path without the remote login.

**Note:** Make sure that the `/tmp/cds_erasefile.<userID>` is deleted from all the remote machines.

#### removeJobData

Specifies whether to remove the results data of a job from the distributed processing database before submitting a new job with the same name. For example, if you submit a job by the name *Job001*, the data for this job is generated in the `<>/..../job001` directory. When you submit another job with the same name, the data for that would be generated at the same location `<>/..../job001`. However, if the `removeJobData` variable is set to `t`, the results of the previous run are deleted before submitting the new job. The GUI option corresponding to the `removeJobData` environment variables is the *Remove Data before job submission* option on the Job Policy Setup form.

To set this variable in the `.cdsenv` file, enter the following:

```
asimenv.distributed removeJobData boolean nil
```

## Environment Variables for Distributed Processing with LBS

This section describes the environment variables that control various distributed processing options. Add the variables you want to use in your `.cdsenv` file.

The following environment variables are available when jobs are distributed using LBS:

- lsfResourceString
- selectLsfResourceString
- drmsCommand
- queueName
- queueNameExclusionList
- queueNameInclusionList
- remoteShell


© 2023 Cadence Design Systems, Inc.

- remoteCopy
- queueName
- lsfApplicationProfile
- selectLsfApplicationProfile
- lsfNoOfProcessors
- selectLsfNoOfProcessors
- sgeNoOfProcessors
- sgeParallelEnvName
- sgeSoftResources
- sgeHardResources
- sgePriority

### lsfResourceString

Specifies the LSF resource requirement string. The resource requirement string is displayed in the LSF Resource String field in the Job Policy Setup form. To use the resource requirement string when submitting a job, select the *LSF Resource String* check box in the Job Policy Setup form, or add the `selectLsfResourceString t` variable. For more information about resource requirement strings, see System Administration of Distributed Jobs with LBS.

To set this variable in the `.cdsenv` file, enter the following:

```
asimenv.distributed lsfResourceString string ""
```

### selectLsfResourceString

If set to nil, the specified LSF resource string is not used when you submit a job, unless you select the *LSF Resource String* check box in the Job Policy Setup form. If set to `t`, the specified LSF resource string is used when a job is submitted.

To set this variable in the `.cdsenv` file, enter the following:

```
asimenv.distributed selectLsfResourceString  boolean  t
```


© 2023 Cadence Design Systems, Inc.

### drmsCommand

Specifies the DRMS commands you want to use to submit jobs in the command mode. For example, you can specify `bsub` commands for LSF or `qsub` commands for SGE (Sun Grid Engine) for submitting jobs using this environment variable. When this environment variable is set, the command option in the Job Policy Setup form is selected by default and the specified DRMS commands are displayed in the command field. For more information about submitting jobs using the command mode, see Submitting Simulation Jobs in LBS Distribution Method.

To set this variable in the `.cdsenv` file, enter the following:

```
asimenv.distributed drmsCommand string ""
```

### queueName

Sets the default queue name. If unspecified, the system default specified in your setup file is used.

To set this variable in the `.cdsenv` file, enter the following:

```
asimenv.distributed queueName  string  ""  nil
```

### queueNameExclusionList

Removes the specified queues from the list of queues available to distributed processing for submitting jobs. For example, if there are four LSF queues named Q1, Q2, Q3, and Q4 and you set the variable to "Q1 Q3", then only Q2 and Q4 will be available to distributed processing for submitting jobs.

To set this variable in the `.cdsenv` file, enter the following:

```
asimenv.distributed queueNameExclusionList string "" nil
```

### queueNameInclusionList

Makes only the specified queues available to distributed processing for submitting jobs. For example, if there are four LSF queues named Q1, Q2, Q3, and Q4 and you set the variable to "Q1 Q4", then only Q1 and Q4 will be available to distributed processing for submitting jobs.

To set this variable in the `.cdsenv` file, enter the following:

```
asimenv.distributed queueNameInclusionList string ""
```


© 2023 Cadence Design Systems, Inc.

### remoteShell

Sets `rsh` as the default remote connection mode and `rcp` as the default remote copy mode for distributed processing. Specify the value `ssh` to use `ssh` as the connection mode and `scp` as the copy mode for distributed processing. You can also specify the path to any other script or command to be used as the default remote connection mode. If you do this, `rcp` will be used as the default remote copy mode. You can use the remoteCopy environment variable to specify a different command to be used as the default remote copy mode.

To set this variable in the `.cdsenv` file, enter the following:

```
asimenv.distributed remoteShell string "rsh"
```

Valid Values: `rsh`, `ssh`, or the path to any other script or command.

Default Values: `rsh`

### remoteCopy

Allows you specify the path to the command to be used as the default remote copy mode for distributed processing. If this environment variable is not set, the default remote copy mode for distributed processing will depend on the settings for the remoteShell environment variable.

To set this variable in the `.cdsenv` file, enter the following:

```
asimenv.distributed remoteCopy string ""
```

### queueName

Specifies name of the queue in which the job is to be submitted. This variable is used in both SGE and LSF modes.

To set this variable in the `.cdsenv` file, enter the following:

```
asimenv.distributed queueName string nil
```

### lsfApplicationProfile

Specifies name of an application profile to be used in LSF mode.

To set this variable in the `.cdsenv` file, enter the following:

```
asimenv.distributed lsfApplicationProfile string
```


© 2023 Cadence Design Systems, Inc.

### selectLsfApplicationProfile

Specifies if the value specified by `lsfApplicationProfile` is to be used in LSF mode.

To set this variable in the `.cdsenv` file, enter the following:

```
asimenv.distributed selectLsfApplicationProfile boolean
```

### lsfNoOfProcessors

Specifies the number of parallel processors to be used in LSF mode.

To set this variable in the `.cdsenv` file, enter the following:

```
asimenv.distributed lsfNoOfProcessors string "3"
```

### selectLsfNoOfProcessors

Specifies if the value specified by lsfNoOfProcessors is to be used in LSF mode.

To set this variable in the `.cdsenv` file, enter the following:

```
asimenv.distributed selectLsfNoOfProcessors boolean
```

### sgeNoOfProcessors

Specifies the default number of processors to be used in SGE mode.

To set this variable in the `.cdsenv` file, enter the following:

```
asimenv.distributed sgeNoOfProcessors string nil
```

### sgeParallelEnvName

Specifies name of the parallel environment in SGE mode.

To set this variable in the `.cdsenv` file, enter the following:

```
asimenv.distributed sgeParallelEnvName string nil
```

### sgeSoftResources

Specifies the default soft resources to be used in SGE mode.

To set this variable in the `.cdsenv` file, enter the following:

```
asimenv.distributed sgeSoftResources string nil
```


© 2023 Cadence Design Systems, Inc.

### sgeHardResources

Specifies the default hard resources to be used in SGE mode.

To set this variable in the `.cdsenv` file, enter the following:

```
asimenv.distributed sgeHardResources string nil
```

### sgePriority

Specifies the default priority to be set for a job submitted in SGE mode. Default value set to 0.

To set this variable in the `.cdsenv` file, enter the following:

```
asimenv.distributed sgePriority int nil
```


© 2023 Cadence Design Systems, Inc.

# adexl.monte

- additionalNetlistOptions
- applySaveOptionsToNetlist
- createStatisticalCornerType
- enableMonteCarloOverStatisticalCorners
- incrementalUpdate
- iterationUpdates
- minGroupSizeSplitAcrossIdleJobs
- numberOfPointsToview
- samplingMethod
- saveProcessOptionDefaultValue
- saveProcessOptionDefaultValue
- saveMismatchOptionDefaultValue
- saveSimulationData


© 2023 Cadence Design Systems, Inc.

## additionalNetlistOptions

`adexl.monte additionalNetlistOptions string` *`any_string`*

### Description

Specifies the additional analysis options to be generated in the netlist.

You cannot specify the `numruns`, `firstrun`, and `seed` options using this environment variable.

The default value is " ".

### GUI Equivalent

| | |
|---|---|
| Command | *Monte Carlo* |
| Field | *Netlist Options* |

### Examples

```
envGetVal("adexl.monte" "additionalNetlistOptions")
envSetVal("adexl.monte" "additionalNetlistOptions" 'string
"nullmfactorcorrelation=yes")
```


© 2023 Cadence Design Systems, Inc.

## applySaveOptionsToNetlist

```
adexl.monte applySaveOptionsToNetlist boolean { t | nil }
```

### Description

Controls the writing of process and mismatch parameter information in the netlist. The valid values are:

- `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 *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.

The default value is `t`.

### Examples

```
envGetVal("adexl.monte" "additionalNetlistOptions")
envSetVal("adexl.monte" "additionalNetlistOptions" 'boolean t)
envSetVal("adexl.monte" "additionalNetlistOptions" 'boolean nil)
```


© 2023 Cadence Design Systems, Inc.

## createStatisticalCornerType

```
adexl.monte createStatisticalCornerType cyclic { sequence | value | prompt |
     promptValues | auto }
```

### Description

Controls the writing of process and mismatch parameter information in the netlist.It has the following valid values:

- `sequece`: Create a statistical corner by using a sequence ID of a sample.
- `value`: 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.
- `auto`: Automatically creates a statistical corner after completion of simulation using the statistical parameter values of an appropriate sample generated by Spectre. The selected sample varies depending on the Monte Carlo method used for running simulation.

The default value is `auto`.

### Examples

```
envGetVal("adexl.monte" "createStatisticalCornerType")
envSetVal("adexl.monte" "createStatisticalCornerType" 'cyclic "auto")
envSetVal("adexl.monte" "createStatisticalCornerType" 'cyclic "sequence")
envSetVal("adexl.monte" "createStatisticalCornerType" 'cyclic "value")
envSetVal("adexl.monte" "createStatisticalCornerType" 'cyclic "prompt")
envSetVal("adexl.monte" "createStatisticalCornerType" 'cyclic "promptValues")
```


© 2023 Cadence Design Systems, Inc.

## enableMonteCarloOverStatisticalCorners

```
adexl.monte enableMonteCarloOverStatisticalCorners boolean { t | nil }
```

### Description

Enables you to run Monte Carlo analysis (mismatch variation) over parameter-based statistical corners, consisting of process variation. It can have the following values:

- `t`: Monte Carlo analysis can be run over parameter-based statistical corners.
- `nil`: Monte Carlo analysis cannot be run with statistical corners.

The default value is `nil`.

### Examples

```
envGetVal("adexl.monte" "enableMonteCarloOverStatisticalCorners")
envSetVal("adexl.monte" "enableMonteCarloOverStatisticalCorners" 'boolean nil)
envSetVal("adexl.monte" "enableMonteCarloOverStatisticalCorners" 'boolean t)
```


© 2023 Cadence Design Systems, Inc.

## incrementalUpdate

```
adexl.monte incrementalUpdate boolean { t | nil }
```

### Description

Controls the update of Monte Carlo simulation results in the Results tab of the Outputs pane. It can have the following 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.

The default value is `t`.

### Examples

```
envGetVal("adexl.monte" "incrementalUpdate")
envSetVal("adexl.monte" "incrementalUpdate" 'boolean t)
envSetVal("adexl.monte" "incrementalUpdate" 'boolean nil)
```


© 2023 Cadence Design Systems, Inc.

## iterationUpdates

```
adexl.monte iterationUpdates int integer
```

### Description

Controls the number of iterations of the Monte Carlo run after which simulation results are updated in the Results tab of the Outputs pane.

The default value is `10`.

### Examples

```
envGetVal("adexl.monte" "iterationUpdates")
envSetVal("adexl.monte" "iterationUpdates" 'int 10)
envSetVal("adexl.monte" "iterationUpdates" 'int 20)
```


© 2023 Cadence Design Systems, Inc.

## minGroupSizeSplitAcrossIdleJobs

```
adexl.monte minGroupSizeSplitAcrossIdleJobs int integer
```

### Description

Specifies the minimum group size to be considered while splitting or distributing the Monte Carlo simulation points to the available idle jobs. By default, the points are not reallocated after an initial assignment to jobs. However, if there are multiple jobs, you can set this variable to a value greater than `1` to enable reallocation of points. In this case, ADE XL identifies the busy or idle jobs, and if the number of pending points on a busy job is more than the specified group size, it reallocates some of the points to the idle jobs, thereby helping in optimum utilization of the available resources.

For example, if a Monte Carlo simulation has 100 points and the *Max Jobs* field on the Job Policy Setup form is set to `2`, a set of 50 simulation points are allocated to each one of `job1` and `job2`. If the simulations running on `job1` are completed faster than those running on `job2`, `job1` becomes idle. If you have specified the minimum size for a group of simulations to be considered for the reallocation of points to 5 and the number of points pending with `job2` is greater than 5, some of the points are reallocated to `job1`. This can improve the overall run time for Monte Carlo simulations.

The default value is `0`.

**Note:** This environment variable is not supported in reliability analysis for Monte Carlo run mode.

### Examples

```
envGetVal("adexl.monte" "minGroupSizeSplitAcrossIdleJobs")
envSetVal("adexl.monte" "minGroupSizeSplitAcrossIdleJobs" 'int 0)
envSetVal("adexl.monte" "minGroupSizeSplitAcrossIdleJobs" 'int 5)
```


© 2023 Cadence Design Systems, Inc.

## numberOfPointsToview

```
adexl.monte numberOfPointsToView int integer
```

### Description

Specifies the number of points to be displayed in the Detail results view for the Monte Carlo Sampling run mode. By default, the Detail view shows all the points of a Monte Carlo run. By setting this variable, you can choose to view only a selected number of worst points.

**Note:** When this environment variable is set to a zero, all the points are displayed in both Detail and Detail - Transpose results views. When it is set to a non-zero value, the Detail - Transpose results view is disabled for the Monte Carlo Sampling run mode.

The default value is `0`.

### Examples

```
envGetVal("adexl.monte" "numberOfPointsToView")
envSetVal("adexl.monte" "numberOfPointsToView" 'int 0)
envSetVal("adexl.monte" "numberOfPointsToView" 'int 5)
```


© 2023 Cadence Design Systems, Inc.

## samplingMethod

```
adexl.monte samplingMethod string lds
```

### Description

Sets the default sampling method for the Monte Carlo Sampling run mode. The valid values are:

- `random`: Random.
- `lhs`: Latin Hypercube Sampling.
- `lds`: Low-Discrepancy Sequence.

The default value is `lds`.

### Examples

```
envGetVal("adexl.monte" "samplingMethod")
envSetVal("adexl.monte" "samplingMethod" 'string "lds")
envSetVal("adexl.monte" "samplingMethod" 'string "random")
envSetVal("adexl.monte" "samplingMethod" 'string "lhs")
```


© 2023 Cadence Design Systems, Inc.

## saveProcessOptionDefaultValue

```
adexl.monte saveProcessOptionDefaultValue boolean { t | nil }
```

### Description

Controls the default setting for the *Save Statistical Parameter Data* check box in the Monte Carlo form.

The valid values are:

- `t`: The *Save Statistical Parameter Data* check box in the Monte Carlo form is selected by default, provided that the environment variable saveMismatchOptionDefaultValue is also set to `t` (if the settings for this option is not already there in the setup).
- `nil`: The *Save Process Data* check box in the Monte Carlo form is deselected by default (if the settings for this option is not already there in the setup).

The default value is `nil`.

### Examples

```
envGetVal("adexl.monte" "saveProcessOptionDefaultValue")
envSetVal("adexl.monte" "saveProcessOptionDefaultValue" 'boolean t)
envSetVal("adexl.monte" "saveProcessOptionDefaultValue" 'boolean nil)
```


© 2023 Cadence Design Systems, Inc.

## saveMismatchOptionDefaultValue

```
adexl.monte saveMismatchOptionDefaultValue boolean { t | nil }
```

### Description

Controls the default setting for the *Save Statistical Parameter Data* check box in the Monte Carlo form.

The valid values are:

- `t`: The *Save Statistical Parameter Data* check box in the Monte Carlo form is selected by default, provided that the environment variable saveProcessOptionDefaultValue is also set to `t` (if the settings for this option is not already there in the setup).
- `nil`: The *Save Process Data* check box in the Monte Carlo form is deselected by default (if the settings for this option is not already there in the setup).

The default value is `nil`.

### Examples

```
envGetVal("adexl.monte" "saveMismatchOptionDefaultValue")
envSetVal("adexl.monte" "saveMismatchOptionDefaultValue" 'boolean t)
envSetVal("adexl.monte" "saveMismatchOptionDefaultValue" 'boolean nil)
```


© 2023 Cadence Design Systems, Inc.

## saveSimulationData

```
adexl.monte saveSimulationData boolean { t | nil }
```

### Description

Specifies if the simulation data is to be saved so that it can be used for plotting. This environment variable sets the default value of the *Save Data To Allow Family Plots* check box in the Monte Carlo form.

The valid values are:

- `t`: The *Save Data To Allow Family Plots* check box in the Monte Carlo form is selected by default.
- `nil`: The *Save Data To Allow Family Plots* check box in the Monte Carlo form is deselected by default.

The default value is `nil`.

### Examples

```
envGetVal("adexl.monte" "saveSimulationData")
envSetVal("adexl.monte" "saveSimulationData" 'boolean nil)
envSetVal("adexl.monte" "saveSimulationData" 'boolean t)
```


© 2023 Cadence Design Systems, Inc.

# AMS


- AMS_IGNORE_IGNORE
- amsIEsList
- hnlVerilogIgnoreTermNameList
- autoConfigNameForConfigCreation
- stopListForConfigCreation
- constraintListForConfigCreation
- dresolution
- finalTimeOp
- infoOptions
- paramRangeCheckFile
- disableCompileVAAsVAMS
- disallowRedefinition
- defaultVsupForVAR
- enablePkgImport



© 2023 Cadence Design Systems, Inc.

# ams.emirOpts

The following table lists the environment variables in the ams.emirOpts partition. These variables work exactly same as the variables in the spectre.emirOpts partition.

| ams.emirOpts environment variables | Corresponding spectre.emirOpts environment variables |
|---|---|
| autoLoadVfiFile | autoLoadVfiFile |
| dspfListBox | dspfListBox |
| editSummaryInfoTable | editSummaryInfoTable |
| layerName | layerName |
| netName | netName |
| short | short |
| shortResLayer | shortResLayer |
| spgsAdvIRDrop | spgsAdvIRDrop |
| spgsFilterMoscap | spgsFilterMoscap |
| techFileMode | techFileMode |


© 2023 Cadence Design Systems, Inc.

# ams.envOpts

## amsIEsList

Specifies the default IE cards and the values of related fields to be loaded in the Interface Element (IE) Setup form when *Interface Element/IE-card Based Setup* is selected. Multiple IE cards can be added to the IE Card table using this variable. The same IE cards and field values are loaded in the form when you click the *Reset all IE Parameters to Default Values* button on the form.

To set this variable in the `.cdsinit` file or CIW, use the call:

```
envSetVal("ams.envOpts" "amsIEsList" 'string "((t \"global\" \"\"
\"Value\" \"1.8\" \"connectLib.CR_full_fast\" \"\" \"logic\"
\"discipline=logic;nox=1;\" \"Built-in\") (t \"inst\" \"/I3\"
\"Value\" \"2.8\" \"connectLib.CR_full_fast\" \"\" \"\"
\"inst=\\\"/I3\\\";\" \"Built-in\"))")
```

| | |
|---|---|
| **Variable Type** | string |
| **Default Value** | "" |


© 2023 Cadence Design Systems, Inc.

**Acceptable Values**

List of lists that specify the following:

- `enableBoolean` — Boolean that specifies whether to enable or disable the IE card.
- `scopeString` — String that specifies the value of the *Scope* field.
- `scopeAppliedToString` — String that specifies the value of the *Scope Applied To* field.
- `vsupNetOrValueString` — String that specifies the value of the *Vsup Value/Net* field.
- `connectRuleNameString` — String that specifies the value of the *connrule* field.
- `modeString` — String that specifies the value of the mode *field*.
- `disciplineString` — String that specifies the value of the *discipline* field
- `updatedParametersString` — String the specifies the value of the *Updated Parameters* field.

**GUI Equivalent**

N/A

## autoConfigNameForConfigCreation

Specifies the default configuration name to be used while creating a config cellview automatically by choosing *Configuration – Auto Create Configuration* from Virtuoso Schematic Editor or Virtuoso Text Editor.

To set this variable in the .cdsinit file or CIW, use the call:

```
envSetVal("ams.envOpts" "autoConfigNameForConfigCreation" 'string
"auto_config")
```

To set this variable in the .cdsenv file, use the call:

```
ams.envOpts autoConfigNameForConfigCreation 'string "auto_config"
```

**Variable Type**

string


© 2023 Cadence Design Systems, Inc.

| | |
|---|---|
| **Default Value** | `auto_config` |
| **Acceptable Values** | Any string value |
| **GUI Equivalent** | N/A |

## case_sensitive

Enables case sensitivity in a regenerated netlist and adds `case_sensitive = true` in the `dspf_include` statement for a Smart View.

For more information, see DSPF Support.

To set this variable in the .cdsinit file or CIW, use the call:

```
envSetVal("ams.envOpts" "case_sensitive" 'string "true")
```

To set this variable in the .cdsenv file, use the call:

```
ams.envOpts case_sensitive 'string "true"
```

| | |
|---|---|
| **Variable Type** | string |
| **Default Value** | `true` |
| **Acceptable Values** | `true`, `false` |
| **GUI Equivalent** | N/A |

## connectRulesList

Sets the default set of connect rules. To set this variable in the .cdsinit file, use the call:

```
envSetVal("ams.envOpts" "connectRulesList" 'string
"connectLib;ConnRules_18V_full;connect
connectLib;mixedsignal;connect")
```

This variable will set two connect rules connectLib/ConnRules_18V_full and connectLib/mixed-signal as the default connect rules in the Connect Rules form.

To set a single connect rule, use the following call:


© 2023 Cadence Design Systems, Inc.

```
envSetVal("ams.envOpts" "connectRulesList" 'string
"connectLib;ConnRules_5V_full;connect")
```

| | |
|---|---|
| **Variable Type** | string |
| **Default Value** | `"connectLib;ConnRules_5V_full;connect"` |

## constraintListForConfigCreation

Specifies the default constraint list to be used while creating a config cellview using Virtuoso Schematic Editor or Virtuoso Text Editor. The same constraint list is used when the config cellview is created automatically. It is also displayed in the *Constraint List* field on the Setup Configuration form.

A constraint list specifies the constraints that apply to the design.

To set this variable in the `.cdsinit` file or CIW, use the call:

```
envSetVal("ams.envOpts" "constraintListForConfigCreation" 'string
"")
```

To set this variable in the `.cdsenv` file, use the call:

```
ams.envOpts constraintListForConfigCreation 'string ""
```


© 2023 Cadence Design Systems, Inc.

| | |
|---|---|
| **Variable Type** | string |
| **Default Value** | `""` |
| **Acceptable values** | A space-separated list of cellview types |

## defaultVsupForVAR

Changes the default value of variable defined in *Supply value/Net.* If the value set in the `defaultVsupForVAR` changes, the value of variable to be defined in the *Supply value/Net* field also changes accordingly.

For example, If the variable set in *Supply value/Net* is `VAR(VDD)` has its value `1.8`, based on the value set for `defaultVsupForVAR`. If the value of `defaultVsupForVAR` is updated, then the value in `VAR(VDD)` is not affected, but the value of next variable that the user sets in *Supply Value/Net* is updated.

To set this variable in the .cdsinit file or CIW, use the call:
```
envSetVal("ams.envOpts" "defaultVsupForVAR" 'string "2.5")
```

| | |
|---|---|
| **Variable Type** | string |
| **Default Value** | 1.8 |
| **Acceptable Values** | `t` |

## disableCompileVAAsVAMS

Controls whether the *Compile VerilogA as Verilog-AMS* field is displayed in the Netlist and Run Options form. If this variable is set to `t`, the field is hidden in the form.

To set this variable in the `.cdsinit` file or CIW, use the call:
```
envSetVal( "ams.envOpts" "disableCompileVAAsVAMS" 'boolean t)
```

To set this variable in the `.cdsenv`, use the following:
```
ams.envOpts disableCompileVAAsVAMS boolean t
```

| | |
|---|---|
| **Variable Type** | boolean |


© 2023 Cadence Design Systems, Inc.

| | |
|---|---|
| **Default Value** | `nil` |
| **Acceptable Values** | `t`, `nil` |

## disableRunModeInDP

When the host mode is distributed processing, setting the value to `t` disables the interactive simulation mode and sets batch simulation mode as default for distributed processing. Setting the value to `nil` will keep both interactive and batch simulation mode active for any host mode.

To set this variable in the `.cdsinit` file or CIW, use the call:
```
envSetVal("ams.envOpts" "disableRunModeInDP" 'boolean t)
```

To set it in the .cdsenv file, add:
```
ams.envOpts disableRunModeInDP 'boolean t
```

| | |
|---|---|
| **Variable Type** | boolean |
| **Default Value** | `nil` |
| **Acceptable Values** | `t, nil` |

## enableChangeSpectrePath

Enables the *Change spectre path* check box in the *Netlist and Run Options* form.

To set this variable in the `.cdsinit` file or CIW, use the call:
```
envSetVal("ams.envOpts" "enableChangeSpectrePath" 'toggle (t
```

To set it in the .cdsenv file, add:
```
ams.envOpts enableChangeSpectrePath 'toggle (t,nil)
```

| | |
|---|---|
| **Variable Type** | boolean |
| **Default Value** | `nil` |
| **Acceptable Values** | `t, nil` |
| **GUI Equivalent** | *Simulation – Netlist and Run Options – Netlist and Run Options form – Change spectre path* |


© 2023 Cadence Design Systems, Inc.

## filesOnIrunCmdLineHDL

This variable is used to specify the files passed to AMS simulator without any options.

To set this variable in the .cdsenv file, add the line
```
ams.envOpts filesOnIrunCmdLineHDL string "((t \"yyy.vams\") (t
\"zzz.vams\"))"
```

To set this variable in the .cdsinit file or CIW, use the call:
```
envSetVal("ams.envOpts" "filesOnIrunCmdLineHDL" 'string "((t
\"yyy.vams\") (t \"zzz.vams\"))")
```

| | |
|---|---|
| **Variable Type** | String |
| **Default Value** | "" |
| **Acceptable Values** | ■ To enable flag — String list of (t \"<file name>\") pair<br>■ To disable flag — String list of (nil \"<file name>\") pair |
| **GUI Equivalent** | *Simulation – Options – AMS Option – Include Option Settings* |

## ipAddIpLabelAsPrefix

If set to t, this variable adds the specified IP Label is added as the prefix to the Virtuoso library name.

To set this variable in the `.cdsinit` file or CIW, use the following call:
```
envSetVal("ams.envOpts" "ipAddIpLabelAsPrefix" 'boolean nil)
```

To set this variable in the `.cdsenv` file, use the following call:
```
ams.envOpts ipAddIpLabelAsPrefix boolean nil
```

| | |
|---|---|
| **Variable Type** | boolean |
| **Default Value** | `nil` |
| **Acceptable Values** | `t`, `nil` |


© 2023 Cadence Design Systems, Inc.

| | |
|---|---|
| **GUI Equivalent** | *AMS IP Export – SystemVerilog Configuration File – Add IP Label to Virtuoso Library Name as Prefix* |

## ipExportHDLFiles

If set to `t`, this variable copies the HDL files to the specified IP Export directory path.

To set this variable in the `.cdsinit` file or CIW, use the following call:

```
envSetVal( "ams.envOpts" "ipExportHDLFiles" 'boolean nil)
```

To set this variable in the `.cdsenv` file, use the following call:

```
ams.envOpts ipExportHDLFiles boolean nil
```

| | |
|---|---|
| **Variable Type** | boolean |
| **Default Value** | `nil` |
| **Acceptable Values** | `t`, `nil` |
| **GUI Equivalent** | *AMS IP Export – Miscellaneous – Export HDL Files* |

## irunIncDirHDL

Specifies a directory to search for the `'include` files.

To set this variable in the `.cdsinit` file or CIW, use the following call:

```
envSetVal( "ams.envOpts" "irunIncDirHDL" 'string "((t
\"/path/include_files/\") (t \".\") )")
```

To set this variable in the `.cdsenv` file, use the following call:

```
ams.envOpts irunIncDirHDL string "((t \"/path/include_files/\") (t
\".\") )"
```

| | |
|---|---|
| **Variable Type** | String |
| **Default Value** | "" |


© 2023 Cadence Design Systems, Inc.

| | |
|---|---|
| **Acceptable Values** | ■ To enable flag — String list of (t \"<directory path>\") pair<br>■ To disable flag — String list of (nil \"<directory path>\") pair |
| **GUI Equivalent** | *Virtuoso Analog Design Environment – Simulation – Options – AMS Option – Include Option Settings* |

## libDirsHDL

Specifies the path to the directory containing the library to be used by AMS simulator.

To set this variable in the `.cdsinit` file or CIW, use the following call:

```
envSetVal( "ams.envOpts" "libDirsHDL" 'string "((t
\"/path/src_files\") (t )")
```

To set this variable in the `.cdsenv` file, use the following call:

```
ams.envOpts libDirsHDL string ""((t \"/path/src_files\") (t )"
```

| | |
|---|---|
| **Variable Type** | String |
| **Default Value** | "" |
| **Acceptable Values** | ■ To enable flag —String list of (t \"<directory path>\") pair<br>■ String list of (nil \"<directory path>\") pair |
| **GUI Equivalent** | *Virtuoso Analog Design Environment – Simulation – Options – AMS Option – Include Option Settings* |

## libFilesHDL

Specifies the name of the file in the library to be used by AMS simulator.

To set this variable in the `.cdsinit` file or CIW, use the following call:

```
envSetVal( "ams.envOpts" "libFilesHDL" 'string "((t
\"module1.vams\") (nil \"/path1/module2.vams\"))")
```


© 2023 Cadence Design Systems, Inc.

To set this variable in the `.cdsenv` file, use the following call:
`ams.envOpts libFilesHDL string "((t \"module1.vams\") (nil \"/path1/module2.vams\"))"`

| | |
|---|---|
| **Variable Type** | String |
| **Default Value** | "" |
| **Acceptable Values** | ■ To enable flag — String list of (t \"<file name>\") pair<br>■ To disable flag — String list of (nil \"<file name>\") pair |
| **GUI Equivalent** | *Virtuoso Analog Design Environment – Simulation – Options – AMS Option – Include Option Settings* |

## libraryListForConfigCreation

Customizes the list of libraries in `.cdsinit` and SKILL programming in CIW. It sets the default global bindings when you set up a configuration or edit an existing configuration using Setup Configuration form or directly by Auto Create Configuration.

To set this variable in the `.cdsinit` file or CIW, use the following call:
`envSetVal("ams.envOpts" "libraryListForConfigCreation" 'string "Lib1 Lib2")`

| | |
|---|---|
| **Variable Type** | String |
| **Default Value** | "" |
| **Acceptable Values** | A single library name, example, `Lib1`. Or a space-separated list of more than one library names, example, `Lib1 Lib2`. |
| **GUI Equivalent** | *Virtuoso Hierarchy Editor Editing– Global Bindings – Library List* |

## maxNumSnapShots

Specifies the maximum number of snapshots to be saved during the specified period for the periodic snapshots.


© 2023 Cadence Design Systems, Inc.

To set this environment variable in the `.cdsinit` file or CIW, use the following call:
`envSetVal("ams.envOpts" "maxNumSnapShots" 'int 4)`

To set this environment variable in the `.cdsenv` file, use the following call:
`ams.envOpts maxNumSnapShots int 4`

| | |
|---|---|
| **Variable Type** | integer |
| **Default Value** | `4` |
| **Acceptable Values** | A positive integer value. |
| **GUI Equivalent** | *Simulation – Netlist and Run Options – SAVE AND RESTART OPTIONS – Periodic snapshot – Max number* |

## optsFileHDL

Specifies the command-line arguments contained in the arguments file for AMS simulator.

To set this variable in the `.cdsinit` file or CIW, use the following call:
`envSetVal("ams.envOpts" "optsFileHDL" 'string "((t \"opt.f\"))")`

To set this variable in the `.cdsenv` file, use the following call:
`ams.envOpts optsFileHDL 'string "((t \"opt.f\"))"`

| | |
|---|---|
| **Variable Type** | string |
| **Default Value** | "" |
| **Acceptable Values** | ■ To enable flag — String list of (t \"<file name>\") pair<br>■ To disable flag — String list of (nil \"<file name>\") pair |
| **GUI Equivalent** | *Virtuoso Analog Design Environment – Simulation – Options – AMS Option – Include Option Settings* |


© 2023 Cadence Design Systems, Inc.

## saveAllSnapShots

Specifies whether to save all snapshots taken during the time window specified for periodic snapshots.

To set this environment variable in the `.cdsinit` file or CIW, use the following call:
`envSetVal("ams.envOpts" "saveAllSnapShots" 'boolean nil)`

To set this environment variable in the `.cdsenv` file, use the following call:
`ams.envOpts saveAllSnapShots boolean nil`

| | |
|---|---|
| **Variable Type** | boolean |
| **Default Value** | `nil` |
| **Acceptable Values** | `t`, `nil` |
| **GUI Equivalent** | *Simulation – Netlist and Run Options – SAVE AND RESTART OPTIONS – Periodic snapshot – Save all snapshots* |

## snapShotBaseName

Specifies the base name of the snapshot.The base name must be alphanumeric with no spaces or special characters.

To set this environment variable in the `.cdsinit` file or CIW, use the following call:
`envSetVal("ams.envOpts" "snapShotBaseName" 'string "mySnapShot")`

To set this environment variable in the `.cdsenv` file, use the following call:
`ams.envOpts snapShotBaseName string "mySnapShot"`

| | |
|---|---|
| **Variable Type** | string |
| **Default Value** | `"mySnapShot"` |
| **Acceptable Values** | An alphanumeric string with no spaces or special characters. |
| **GUI Equivalent** | *Simulation – Netlist and Run Options – SAVE AND RESTART OPTIONS – Snapshot base name* |


© 2023 Cadence Design Systems, Inc.

## snapShotNameTimeUnit

Specifies the time unit to be used in the snapshot name.

For example, if you specify the *Strobe time(s)* as `20n`, and the *Snapshot name time unit* as `us`, a snapshot named `mySnapShot_20n_us_run1` is saved.

**Note:** This variable can only be used if the *Save all snapshots* check box is selected.

To set this environment variable in the `.cdsinit` file or CIW, use the following call:
`envSetVal("ams.envOpts" "snapShotNameTimeUnit" 'string "us")`

To set this environment variable in the `.cdsenv` file, use the following call:
`ams.envOpts snapShotNameTimeUnit string "us"`

| | |
|---|---|
| **Variable Type** | string |
| **Default Value** | `us` |
| **Acceptable Values** | `ps`, `ns`, `us`, `ms`, `s` |
| **GUI Equivalent** | *Simulation – Netlist and Run Options – SAVE AND RESTART OPTIONS – Snapshot time unit* |

## snapShotSaveMode

Specifies the mode for saving the snapshots. The two modes are *Time point*s and *Periodic snapshot*.

To set this environment variable in the `.cdsinit` file or CIW, use the following call:
`envSetVal("ams.envOpts" "snapShotSaveMode" 'string "Time points")`

To set this environment variable in the `.cdsenv` file, use the following call:
`ams.envOpts snapShotSaveMode string "Time points"`

| | |
|---|---|
| **Variable Type** | string |
| **Default Value** | `"Time points"` |
| **Acceptable Values** | `Time point`, `Periodic snapshot` |
| **GUI Equivalent** | *Simulation – Netlist and Run Options – SAVE AND RESTART OPTIONS – Save snapshot* |


© 2023 Cadence Design Systems, Inc.

## spectrePath

Allows to specify a different version of Spectre when using the AMSD flex flow.

**Note the following:**

- This environment variable can be used only if the *Enable AMS flexible release matrix* check box in the *Netlist and Run Options* form is selected and enableChangeSpectrePath is set to `t`.
- It is recommended to use the latest version of Xcelium and Spectre while using the AMSD flex flow.

To set this environment variable in the `.cdsinit` file or CIW, use the following call:
```
envSetVal("ams.envOpts" "spectrePath" 'string "")
```

To set this environment variable in the `.cdsenv` file, use the following call:
```
ams.envOpts spectrePath 'string ""
```

| | |
|---|---|
| **Variable Type** | string |
| **Default Value** | `""` |
| **Acceptable Values** | A valid path to a Spectre build. |
| **GUI Equivalent** | *Simulation – Netlist and Run Options – Netlist and Run Options form – Change spectre path* |

## stopListForConfigCreation

Specifies the default stop list to be used while creating a config cellview using Virtuoso Schematic Editor or Virtuoso Text Editor. The same stop list is used when the config cellview is created automatically. It is also displayed in the *Stop List* field on the Setup Configuration form.

A stop list is a list of views that are to be treated as leaf nodes, that is, not to be expanded.

To set this variable in the `.cdsinit` file or CIW, use the call:
```
envSetVal("ams.envOpts" "stopListForConfigCreation" 'string
"spectre")
```


© 2023 Cadence Design Systems, Inc.

To set it in the `.cdsenv` file, add:
`ams.envOpts stopListForConfigCreation 'string "spectre"`

| | |
|---|---|
| **Variable Type** | string |
| **Default Value** | `spectre` |
| **Acceptable Values** | A space-separated list of cellview types |

## strobeTime

Specifies the time interval between saving snapshots, followed by the time unit.

To set this environment variable in the `.cdsinit` file or CIW, use the following call:
`envSetVal("ams.envOpts" "strobeTime" 'string "")`

To set this environment variable in the `.cdsenv` file, use the following call:
`ams.envOpts strobeTime string ""`

| | |
|---|---|
| **Variable Type** | string |
| **Default Value** | `""` |
| **Acceptable Values** | Any integer value greater than 0, followed by the time unit. |
| **GUI Equivalent** | *Simulation – Netlist and Run Options – SAVE AND RESTART OPTIONS – Strobe time(s)* |

## useleSetup

If this variable is set to `t`, the Interface Element/IE-card based setup is selected for AMS simulation. By default, the Connect Rule/Connect Module based setup is selected for AMS simulation.

After setting this variable to `t`, when you open the Interface Element (IE) setup form, the *Interface Element/IE-card Based Setup(OSS/UNL)* radio button is selected and the additional options required for the Interface Element/IE-card based setup are displayed in the form. Also, when you netlist the design while the Interface Element/IE-card based setup is selected, an `ie_card.scs` file is placed in the netlist directory.


© 2023 Cadence Design Systems, Inc.

The `ie_card.scs` file contains information about how the connect rules are used during simulation.

To set this variable in the `.cdsenv` file, use the call:
`ams.envOpts useIeSetup boolean t`

To set this variable in the `.cdsinit` file or CIW, use the call:
`envSetVal("ams.envOpts" "useIeSetup" 'boolean t)`

| | |
|---|---|
| **Variable Type** | boolean |
| **Default Value** | `nil` |
| **Acceptable Values** | `t`, `nil` |

## viewListForConfigCreation

Specifies the default view list to be used while creating a config cellview using Virtuoso Schematic Editor or Virtuoso Text Editor. The same view list is used when the config cellview is created automatically. It is also displayed in the *View List* field on the Setup Configuration form.

A view list specifies the views to be added in your configuration. It applies to every level of the configuration and determines which view is selected for each object in the design, unless overridden by a cell or instance binding. You can use the asterisk character (`*`) as a wildcard in the view list.

To set this variable in the `.cdsinit` file or CIW, use the call:
`envSetVal("ams.envOpts" "viewListForConfigCreation" 'string "spectre spice pspice verilog verilogams behavioral functional systemVerilog schematic veriloga vhdl vhdlams wreal")`

To set it in the `.cdsenv` file, add:
`ams.envOpts viewListForConfigCreation 'string "spectre spice pspice verilog verilogams behavioral functional systemVerilog schematic veriloga vhdl vhdlams wreal"`

| | |
|---|---|
| **Variable Type** | string |
| **Default Value** | `spectre spice pspice verilog verilogams behavioral functional systemVerilog schematic veriloga vhdl vhdlams wreal` |


© 2023 Cadence Design Systems, Inc.

| | |
|---|---|
| **Acceptable Values** | A space-separated list of cellview types |

## switchViewList

Specifies the default view list. It is a list of the views for ADE to switch when searching for design variables and when netlisting. The same view list is displayed in the *Switch View List* field on the Environment Options form.

To set this variable in the .cdsinit file or CIW, use the call:
```
envSetVal("spectre.envOpts" "switchViewList" 'string "schematic spectre")
```

To set this variable in the .cdsenv file, use the call:
```
spectre.envOpts switchViewList 'string "schematic spectre"
```

| | |
|---|---|
| **Variable Type** | string |
| **Default Value** | `spectre cmos_sch cmos.sch schematic veriloga` |
| **Acceptable Values** | View names, separated by spaces. |
| **GUI Equivalent** | *Virtuoso Analog Design Environment* – Environment Options – *Switch View List.* |

## enableAMSdebug

Selects or deselects the *Enable mixed signal debugger* check box in the Netlist and Run Options form. This launches the SimVision debug environment for AMS simulator.

To set this variable in the .cdsinit file or CIW, use the call:
```
envSetVal("ams.envOpts" "enableAMSdebug" 'toggle (nil))
```

To set this variable in the .cdsenv file, use the call:
```
ams.envOpts enableAMSdebug toggle (nil)
```

| | |
|---|---|
| **Variable Type** | Boolean |
| **Default Value** | `(nil)` |


© 2023 Cadence Design Systems, Inc.

| | |
|---|---|
| **Acceptable Values** | `(t, nil)` |
| **GUI Equivalent** | *Simulation – Netlist and Run Options form – Enable mixed signal debugger* |

## ipPreserveEnvVars

Selects or deselects the *Preserve Env Variables* check box in the AMS IP Export form.

To set this variable in the .cdsinit file or CIW, use the call:
```
envSetVal("ams.envOpts" "ipPreserveEnvVars" 'boolean nil)
```

To set this variable in the .cdsenv file, use the call:
```
ams.envOpts ipPreserveEnvVars boolean nil
```

| | |
|---|---|
| **Variable Type** | boolean |
| **Default Value** | `nil` |
| **Acceptable Values** | `t, nil` |
| **GUI Equivalent** | *AMS IP Export form – Miscellaneous Tab – Preserve Env Variables* |

## ipEnableDutCell

Selects or deselects the *Enable Export DUT Cell As Top* check box in the AMS IP Export form.

To set this variable in the .cdsinit file or CIW, use the call:
```
envSetVal("ams.envOpts" "ipEnableDutCell" 'boolean nil)
```

To set this variable in the .cdsenv file, use the call:
```
ams.envOpts ipEnableDutCell boolean nil
```

| | |
|---|---|
| **Variable Type** | boolean |
| **Default Value** | `nil` |
| **Acceptable Values** | `t, nil` |


© 2023 Cadence Design Systems, Inc.

| | |
|---|---|
| **GUI Equivalent** | *AMS IP Export form – Miscellaneous Tab – Enable Export DUT Cell As Top* |

## ipDutCell

Specifies the DUT cell name in the *DUT Configuration* group box in the *Miscellaneous* tab of the AMS IP Export form.

To set this variable in the .cdsinit file or CIW, use the call:
```
envSetVal("ams.envOpts" "ipDutCell" 'string " ")
```

To set this variable in the .cdsenv file, use the call:
```
ams.envOpts ipEnableDutCell string " "
```

| | |
|---|---|
| **Variable Type** | string |
| **Default Value** | `""` |
| **Acceptable Values** | Name of the cell from where you want to export the IP block. |
| **GUI Equivalent** | *AMS IP Export form – Miscellaneous Tab –DUT Configuration – Cell* |

## ipDutLib

Specifies the DUT library name in the *DUT Configuration* group box in the *Miscellaneous* tab of the AMS IP Export form.

To set this variable in the .cdsinit file or CIW, use the call:
```
envSetVal("ams.envOpts" "ipDutLib" 'string " ")
```

To set this variable in the .cdsenv file, use the call:
```
ams.envOpts ipDutLib string " "
```

| | |
|---|---|
| **Variable Type** | string |
| **Default Value** | `""` |
| **Acceptable Values** | Name of the library from where you want to export the IP block. |


© 2023 Cadence Design Systems, Inc.

| | |
|---|---|
| **GUI Equivalent** | *AMS IP Export form – Miscellaneous Tab –DUT Configuration – Library* |

## ipDutView

Specifies the DUT library name in the *DUT Configuration* group box in the *Miscellaneous* tab of the AMS IP Export form.

To set this variable in the .cdsinit file or CIW, use the call:

```
envSetVal("ams.envOpts" "ipDutView" 'string " ")
```

To set this variable in the .cdsenv file, use the call:

```
ams.envOpts ipDutView string " "
```

| | |
|---|---|
| **Variable Type** | string |
| **Default Value** | `""` |
| **Acceptable Values** | Name of the view from where you want to export the IP block. |
| **GUI Equivalent** | *AMS IP Export form – Miscellaneous Tab –DUT Configuration – View* |


© 2023 Cadence Design Systems, Inc.

## ieUseUcmAsDefault

```
ams.envOpts ieUseUcmAsDefault boolean { t| nil}
```

### Description

Selects or deselects the *Use UCM as Default* check box in the *Interface Element (IE) Setup* form. This check box specifies that by default, any new IE card must be created by using the Universal Connect Module (UCM).

The default value is `nil`.

### GUI Equivalent

| | |
|---|---|
| Command | *Setup – Connect Rules/IE Setup– Interface Element (IE) Setup* |
| Field | *Use UCM as Default* |

### Examples

```
envGetVal("ams.envOpts" "ieUseUcmAsDefault")
```

```
envSetVal("ams.envOpts" "ieUseUcmAsDefault" 'boolean t)
```


© 2023 Cadence Design Systems, Inc.

# ams.netlisterOpts

## unlAnalogOnly

```
ams.netlisterOpts unlAnalogOnly string { Pure Analog Cellviews| Extracted Views|
     ""}
```

### Description

Specifies the value for the option *Netlist using Spectre* in the *Netlister* tab of the AMS Options form.

- `""`: Disables netlisting using Spectre.

  This is the default.
- `Pure Analog Cellviews`: Enables Spectre netlisting for pure analog cellviews.
- `Extracted Views`: Enables Spectre netlisting for extracted views.

### GUI Equivalent

| | |
|---|---|
| Command | *Simulation – Options – AMS Simulator* |
| Field | *Netlist using Spectre* |

### Examples

```
envGetVal("ams.netlisterOpts" "unlAnalogOnly")
```

```
envSetVal("ams.netlisterOpts" "unlAnalogOnly" 'string "Pure Analog Cellviews")
```


© 2023 Cadence Design Systems, Inc.

# asimenv

## ignoreUnsetDesVars

Ignores the design variables with undefined values, while running a simulation and disables the warning message.

To set this variable in the .cdsinit file or CIW, use the call:
`envSetVal("asimenv" "ignoreUnsetDesVars" 'boolean nil))`

To set this variable in the .cdsenv file, use the call:
`asimenv enableAMSdebug boolean nil`

| | |
|---|---|
| **Variable Type** | Boolean |
| **Default Value** | `nil` |
| **Acceptable Values** | `t`, `nil` |


© 2023 Cadence Design Systems, Inc.

# asimenv.monte

## highPrecision

Controls the numerical precision of the results in the *Results Display Window*. This window gets displayed on using the forms, *Correlation Table* and *Plot/Print Versus Iteration*.

By default, the results displayed are accurate up to 4 decimal places.

If set to `t`, the results displayed are accurate up to 6 decimal places.

To set this variable in the `.cdsinit` file or CIW, use the call:

```
envSetVal("asimenv.monte" "highPrecision" 'boolean nil)
```

To set it in the `.cdsenv` file, add:

```
asimenv.monte highPrecision 'boolean nil
```

| | |
|---|---|
| **Variable Type** | Boolean |
| **Default Value** | `nil` |
| **Acceptable Values** | ■ `t` –The results displayed are accurate up to 6 decimal places.<br>■ `nil` – The results displayed are accurate up to 4 decimal places. |

## precisionDigits

Overrides the numerical accuracy specified through the variable highPrecision.

To set this variable in the `.cdsinit` file or CIW, use the call:

```
envSetVal("asimenv.monte" "precisionDigits" 'int 0)
```

To set it in the `.cdsenv` file, add:

```
asimenv.monte precisionDigits int 0
```

| | |
|---|---|
| **Variable Type** | Integer |
| **Default Value** | `0` |
| **Acceptable Values** | A positive integer value. |


© 2023 Cadence Design Systems, Inc.

# asimenv.netlist

## issueInfoMsgOnCellRename

```
asimenv.netlist issueInfoMsgOnCellRename boolean { t | nil }
```

### Description

Prints information messages when a subcircuit is renamed due to a conflict.

For example, if the same instance comes from two different libraries in one schematic view, the subcircuit is renamed and an information message is displayed.

The default is `nil`.

### GUI Equivalent

None

### Examples

```
envGetVal("asimenv.netlist" "issueInfoMsgOnCellRename")
envSetVal("asimenv.netlist" "issueInfoMsgOnCellRename" 'boolean t)
```


© 2023 Cadence Design Systems, Inc.

# auCore.misc

## doExactTermMatch

```
auCore.misc doExactTermMatch boolean { t | nil }
```

### Description

Specifies whether the tool compares the terminal order in the cellview with the terminal order in CDF during check and save. By default, this variable is set to nil and ADE Explorer or ADE Assembler ignore the order of terminals. This lets you ignore terminal order mismatches caused by the the format in which terminals are saved.

### GUI Equivalent

None

### Examples

```
envGetVal("auCore.misc" "doExactTermMatch")
envSetVal("auCore.misc" "doExactTermMatch" 'boolean t)
```


© 2023 Cadence Design Systems, Inc.

# auCore.selection

## alwaysFlattenNames

```
auCore.selection alwaysFlattenNames boolean { t | nil }
```

### Description

Specifies whether to display or hide the *Include as bundle* check box on the Select bits from bus form.

The default value is `nil`.

### GUI Equivalent

The *Include as bundle* check box in the Select bits from bus form.

### Examples

```
envGetVal("auCore.selection" "alwaysFlattenNames")
```

```
envSetVal("auCore.selection" "alwaysFlattenNames" 'boolean t)
```


© 2023 Cadence Design Systems, Inc.

# HspiceD

- hspiceMaxLineLength
- hspiceSoftLineLength
- mapGndNetToZero
- netlistModelFileFirst
- setTopLevelAsSubckt
- userCmdLineOption


© 2023 Cadence Design Systems, Inc.

# license

- AMSEnvLicenseCheckoutOrder
- maestroCheckoutOrder


© 2023 Cadence Design Systems, Inc.

# maestro.cpupdtr

### keepHistoryData

Specifies whether or not to copy the history of a maestro view while copying the tests in the view.

If you do not wish to copy the simulation results, it is recommended to also set the variable, `copyResultsData` in your `.cdsenv` file and set its value to `nil`.

**Note:** The `keepHistoryData` variable is used by both Virtuoso and a supporting utility. Therefore, it is recommended to set it in the `~/.cdsenv` file that is read by both. If you set this variable in the `<current-working-directory>/.cdsenv` file, you must also set the `CDS_LOAD_ENV` variable in the UNIX environment to CSF. The variable is not used when specified in the `.cdsinit` file.

To set this variable in .cdsenv:

```
maestro.cpupdtr keepHistoryData boolean t
```

| | |
|---|---|
| **Variable Type** | boolean |
| **Default Value** | t |
| **Acceptable Values** | t, nil |


© 2023 Cadence Design Systems, Inc.

# maestro.distribute

### spectreMTNumberUsed

Specifies the value of the `+mt` option for Spectre APS when on the High-Performance Simulation Options form, the *Manual* multi-threading mode option is selected for APS Spectre simulation performance.

To set this variable in the .cdsenv:

```
maestro.distribute spectreMTNumberUsed cyclic "Estimated"
```

In .cdsinit or the CIW:

```
envSetVal( "maestro.distribute" "spectreMTNumberUsed" 'cyclic
"Estimated")
```

| | |
|---|---|
| **Variable Type** | cyclic |
| **Default Value** | `Estimated` |
| **Acceptable Values** | ■ `"Specified"`: `+mt` is set to the value of the *#Threads* field on the High-Performance Simulation Options form.<br>■ `"Estimated"`: If the resource estimation for the current job policy is enabled, `+mt` is set to the thread count estimated for the job policy. Otherwise, `+mt` is set to the value of the *#Threads* field.<br>**Note:** You can enable resource estimation for the current job policy by using the options given in the Provide CPU and Memory Data group box on the Resources tab of the Job Policy Setup form.<br>■ `"Automatic"`: If the resource estimation for the current job policy is enabled, compares the thread count estimated for the current job policy with value of the *#Threads* field, and uses the smaller value. Otherwise, `+mt` is set to the value of the *#Threads* field. |


© 2023 Cadence Design Systems, Inc.

# maestro.emir

## showEnableRFReferenceFlowUI

```
maestro.emir showEnableRFReferenceFlowUI boolean {t | nil}
```

### Description

Selects the *Enable RF Reference Flow* check box in the *Basic* tab of the EMIR Analysis Setup form that enables the RF reference flow. This lets you add passive devices for electromigration analysis like inductors or transformers.

The default value is `nil`.

### GUI Equivalent

| | |
|---|---|
| Command: | *Setup – EMIR Analysis* |
| Field: | *Enable RF Reference Flow* |

### Examples

```
envGetVal("maestro.emir" "showEnableRFReferenceFlowUI")
envSetVal("maestro.emir" "showEnableRFReferenceFlowUI" 'boolean t)
```


© 2023 Cadence Design Systems, Inc.

# maestro.explorer

### defaultOutputColumns

Specifies the default list of columns to be shown on the *Outputs Setup* tab in ADE Explorer.

To set this variable in the .cdsenv:

```
maestro.explorer defaultOutputColumns 'string "\"Name\" \"Type\"
\"Details\" \"Value\" \"Plot\" \"Save\" \"Spec\"
```

In .cdsinit or the CIW:

```
envSetVal( "maestro.explorer" "defaultOutputColumns" 'string
"\"Name\" \"Type\" \"Details\" \"Value\" \"Plot\" \"Save\" \"Spec\"")
```

| | |
|---|---|
| **Variable Type** | string |
| **Default Value** | "\"Name\" \"Type\" \"Details\" \"Value\" \"Plot\" \"Save\" \"Spec\"" |
| **Acceptable Values** | Valid column names, including user-defined column names |

### displayImplicitSignals

Controls whether to show or hide the implicit signals in ADE Explorer. While creating expressions in the *Details* field, if it uses a signal or net that does not already exist in the outputs table, a new row for an output of type `Implicit` is added to the table. For example, when you create an output expression, `average("/net06"))`, an implicit output is created for signal `/net06`.

By default, implicit signals are not shown in the outputs table.

To set this variable in .cdsinit or CIW:

```
envSetVal("maestro.explorer" "displayImplicitSignals" 'boolean nil)
```


© 2023 Cadence Design Systems, Inc.

| | |
|---|---|
| **Variable Type** | boolean |
| **Default Value** | `nil` |
| **Acceptable Values** | ■ `t` — Displays implicit signals in ADE Explorer<br>■ `nil` — Hides implicit signals in ADE Explorer |
| **GUI Equivalent** | *Results — Implicit Signals* |

### outputSetupFilteringNumOutputThreshold

Sets the minimum limit after which filters start appearing on the columns in the Output Setup pane in ADE Explorer. When you set the value of this variable to `0`, filters will always appear regardless of the number of output rows. On the other hand, if you set the value to a large number, say `99999`, filters will never appear.

You can use these filters to search and filter results based on the specified text. For more information about how to perform filtering in Outputs Setup pane, see Filtering and Searching in Outputs Setup Pane on page 462.

To set this variable in the .cdsinit file or CIW:

```
envSetVal("maestro.explorer"
"outputSetupFilteringNumOutputThreshold" 'int 20)
```

| | |
|---|---|
| **Variable Type** | integer |
| **Default Value** | 20 |
| **Acceptable Values** | from 0 to 99999 |

### retainNetlistDirForSinglePoint

Determines whether the netlist directory is to be retained when the next simulation is a single-point run. When this variable is set to `t` (default value), the netlist directory is retained for the next simulation run. If set to `nil`, the netlist directory is deleted before the next simulation run starts.

To set this variable in the .cdsinit file or CIW:


© 2023 Cadence Design Systems, Inc.

```
envSetVal("maestro.explorer" "retainNetlistDirForSinglePoint"
'boolean t)
```

| | |
|---|---|
| **Variable Type** | boolean |
| **Default Value** | t |
| **Acceptable Values** | t, nil |

### showSaveBySubcktAssistant

Controls the display of Save By Subckt Instances assistant in ADE Explorer. The default value of this variable is `auto`, which means this assistant is displayed only when the saved subcircuit information is available. When set to `always`, the Save By Subckt Instances assistant is always displayed in ADE Explorer and schematic. When set to `never`, this assistant is not displayed.

However, if you open Save By Subckt Instances assistant using the *Outputs – To Be Saved – Select By Subckt Inst* menu, it always appears regardless of the setting of this variable.

To set this variable in the .cdsinit file or CIW:

```
envSetVal("maestro.explorer" "showSaveBySubcktAssistant" 'cyclic
"auto")
```

| | |
|---|---|
| **Variable Type** | string |
| **Default Value** | auto |
| **Acceptable Values** | auto, never, always |

### showWhatsNewWindow

Controls whether the What's New window should be displayed when ADE Explorer is opened for the first time. This window highlights the new features in ADE Explorer and is displayed only in those releases that include significant new features. If a release does not include a new feature, the What's New window is not displayed regardless of the setting of this environment variable. In addition, the value of this environment variable is always in synchronization with the *Do not show this again* check box on the What's New window.


© 2023 Cadence Design Systems, Inc.

For example, suppose you are using the `IC6.1.7ISR3` or `ICADV12.2ISR3` release and this window is enabled for these releases. Now, if you select the *Do not show this again* check box, this window will not appear next time when you open ADE Explorer in these releases. Also, the value of variable is automatically set to the specific release number. In this example, it is set to `IC6.1.7-64b.500.3` or `ICADV12.2-64b.500.3`. Alternatively, to stop displaying the What's New window for a specific release, instead of clicking the *Do not show this again* check box, you can also manually set the value of this variable to the specific release number.

Now, if you move to `IC6.1.7ISR4` or `ICADV12.2ISR4`,

- If the What's New window is enabled, the value of this variable and the `Do not show this again` check box are reset, and the What's New Window is always displayed to highlight the new features.
- If the What's New window is not enabled, it will not be displayed.

To set this variable in the .cdsinit file or CIW:

```
envSetVal("maestro.explorer" "showWhatsNewWindow" 'string "yes")
```

| | |
|---|---|
| **Variable Type** | string |
| **Default Value** | yes |
| **Acceptable Values** | any string value indicating a release number |


© 2023 Cadence Design Systems, Inc.

# maestro.gui

## confirmBeforeTestOrCornerDelete

Controls the display of the confirmation message that appears when you delete a test or a corner in the Setup assistant of ADE Explorer or in the Data View assistant of ADE Assembler. When this variable is set to `t`, the confirmation message appears every time you delete a test or a corner. When this variable is set to `nil`, which is the default value, the confirmation message will be disabled for the current Virtuoso session.

To set this variable in the .cdsinit file or CIW:

```
envSetVal("maestro.gui" "confirmBeforeTestOrCornerDelete" 'boolean
nil)
```

| | |
|---|---|
| **Variable Type** | boolean |
| **Default Value** | nil |
| **Acceptable Values** | t, nil |

## nameDisplayWidthInDataView

Controls the width of the divider bar that appears on top of the Data View and Setup assistants. You can drag this bar to resize the width of the Name and Value columns.

This variable can be set to the following values:

- `Fixed`—Divider bar is fixed. You cannot drag the bar to resize the column width.
- `FitToColumn`—Column width is automatically adjusted according to the length of the names or values in the columns.
- `Interactive`—Divider bar is moveable. You can drag the bar to change the column width.
- `UseVariableNameWidth`—At first, the column width is automatically adjusted according to the length of the names or values in the columns. Later, you can drag the bar to change the column width. For example, if a new variable is added and it is longer than the longest column name, then the slider will have to be manually adjusted to see the new column name.

To set this variable in the .cdsinit file or CIW:


© 2023 Cadence Design Systems, Inc.

```
envSetVal("maestro.gui" "nameDisplayWidthInDataView" 'cyclic
"Interactive")
```

| | |
|---|---|
| **Variable Type** | cyclic |
| **Default Value** | Interactive |
| **Acceptable Values** | Fixed, FitToColumn, Interactive, UseVariableNameWidth |

## resultsFilteringColumnsList

Specifies the threshold number of rows after which the column filters are displayed on the Results tab of ADE Explorer and ADE Assembler. By default, the variable is set to `0`. Therefore, the column filters are always displayed. Set this variable to a larger number to hide the filters for a small number of rows.

In .cdsenv:

```
maestro.gui resultsFilteringColumnsList int 0
```

In .cdsinit or the CIW:

```
envSetVal( "maestro.gui" "resultsFilteringColumnsList" 'int 0)
```

| | |
|---|---|
| **Valid Values** | Positive integer value<br>The threshold number of rows after which column filters are displayed. |
| **Default Value** | 0 |

## showFilterInDataView

Determines whether to display filters on top of the *Name* and *Value* columns in the Setup assistant. When this variable is set to `t`, which is the default value, the filters are shown. When set to `nil`, the filters are hidden. You can use these filters to filter the column contents of the Setup assistant.


© 2023 Cadence Design Systems, Inc.

**Note:** The setting of this variable is applicable only when ADE Explorer is opened for the first time. You can also change this setting by right-clicking the column header in the Setup assistant and choosing *Show Filters* or *Hide Filters*. When you change the setting using user interface, it will always be considered for the given session regardless of the setting of this variable.

To set this variable in the .cdsinit file or CIW:

```
envSetVal("maestro.gui" "showFilterInDataView" 'boolean "t")
```

| | |
|---|---|
| **Variable Type** | boolean |
| **Default Value** | t |
| **Acceptable Values** | t, nil |

## useExpressionBuilder

Determines whether to use the Expression Builder or the Calculator window to create output expressions in the ADE Outputs Setup pane. When this variable is set to `t`, the Expression Builder opens up when you create expressions. When it is set to `nil`, the Calculator window appears that you can use to build expressions.

To set this variable in the .cdsinit file or CIW:

```
envSetVal("maestro.gui" "useExpressionBuilder" 'boolean nil)
```

| | |
|---|---|
| **Variable Type** | boolean |
| **Default Value** | t |
| **Acceptable Values** | t, nil |

## usePlottingTemplate

Controls the display of the Plotting Template drop-down list in ADE Explorer and ADE Assembler.

To set this variable in the .cdsinit file or CIW:


© 2023 Cadence Design Systems, Inc.

```
envSetVal("maestro.gui" "usePlottingTemplate" 'boolean t)
```

| | |
|---|---|
| **Variable Type** | boolean |
| **Default Value** | `t` |
| **Acceptable Values** | `t`, `nil` |

## yieldViewMaxOutputsShowExpanded

Specifies the threshold value for the number of output rows that can be displayed in expanded form in the Yield results view. If the output rows exceed this threshold, the results are displayed in collapsed form.

To set this variable in the .cdsinit file or CIW:

```
envSetVal( "maestro.gui" "yieldViewMaxOutputsShowExpanded" 'int 20)
```

| | |
|---|---|
| **Variable Type** | integer |
| **Default Value** | 20 |
| **Acceptable Values** | A positive integer |

## saveImplicitSignalsDuringRun

Ensures that the implicit signals are added to the `save` statement in the netlist during the simulation run in the following scenarios:

- The output expressions, containing implicit signals, were created using ADE Explorer or Assembler (IC6.1.8 ISR5 version or below). Simulation is not run using these expressions.


© 2023 Cadence Design Systems, Inc.

- The *Select signals to output (save)* option on the Save options form is set to *selected* or *allpub* during the creation of an output expression.

To set this variable in .cdsinit or CIW:

```
envSetVal("maestro.gui" "saveImplicitSignalsDuringRun" 'boolean nil)
```

| | |
|---|---|
| **Variable Type** | boolean |
| **Default Value** | nil |
| **Acceptable Values** | ■ `t` — Saves implicit signals while running a simulation<br>■ `nil` — Does not save implicit signals while running a simulation |
| **GUI Equivalent** | N/A |

## hideShowMultiProcess

Controls whether to show or hide the option, *Multi-processing,* in the General section of High-Performance Simulation Options form, when Simulation performance mode is set to *Spectre X*.

To set this variable in .cdsinit or CIW:

```
envSetVal("maestro.gui" "hideShowMultiProcess" 'boolean t)
```


© 2023 Cadence Design Systems, Inc.

| | |
|---|---|
| **Variable Type** | boolean |
| **Default Value** | `nil` |
| **Acceptable Values** | ■ `t` — Hides the option *Multi-processing* in the General section of High-Performance Simulation Options form<br>■ `nil` — Displays the option *Multi-processing* in the General section of High-Performance Simulation Options form |
| **GUI Equivalent** | *High Performance Simulation Options — Multi-Processing* |

## showUnitsInResults

Controls whether to hide or show units, such as `Hz` or `dB`,in the measured results for the expressions in the Results tab.

To set this variable in the .cdsenv:

```
maestro.gui showUnitsInResults boolean t
```

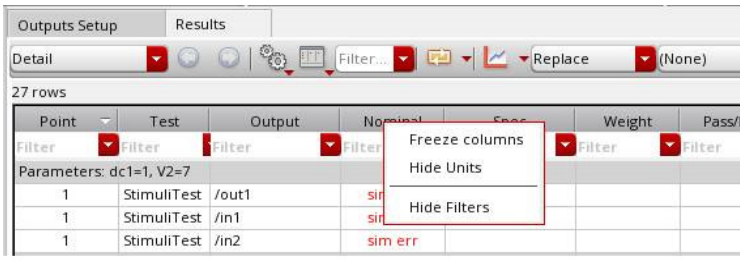
In .cdsinit:

```
envSetVal("maestro.gui" "showUnitsInResults" 'boolean t)
```

| | |
|---|---|
| **Variable Type** | boolean |
| **Default Value** | `t` |
| **Acceptable Values** | ■ `t` — Displays the unit values in the Results tab<br>■ `nil` — Hides the unit in the Results tab |


© 2023 Cadence Design Systems, Inc.

**GUI Equivalent** Right-click menu in the Results column header




## textColorForSpecFailInResults

Specifies a text color, color name, or a color value in hexadecimal format, to be used for the *Pass/Fail*, *Min*, and *Max* columns for the result values that fail to meet the specification.

To set this variable in .cdsinit or CIW:

```
maestro.gui textColorForSpecFailInResults "colorName_or_hexadecimalValue"
```

| | |
|---|---|
| **Variable Type** | string |
| **Default Value** | "#ff0000" |
| **Acceptable Values** | Color name or hexadecimal value |
| **GUI Equivalent** | N/A |

### Example

```
envSetVal("maestro.gui" "textColorForSpecFailInResults" 'string "maroon")
```

Sets the color to be used for values that fail the spec results to maroon, as shown below.

Note that in this example, the *Max* value for output UGF was meeting the spec. Therefore, only the pass/fail status and the *Min* value are in the color specified by this env var.


© 2023 Cadence Design Systems, Inc.

## textColorForSpecNearInResults

Specifies a text color, color name, or a color value in hexadecimal format, to be used for the *Pass/Fail*, *Min*, and *Max* columns for the result values that are near the specification.

To set this variable in .cdsinit or CIW:

```
maestro.gui textColorForSpecNearInResults "colorName_or_hexadecimalValue"
```

| | |
|---|---|
| **Variable Type** | string |
| **Default Value** | "#D2691E" |
| **Acceptable Values** | Color name or hexadecimal value |
| **GUI Equivalent** | N/A |

### Example

```
envSetVal("maestro.gui" "textColorForSpecNearInResults" 'string "#AC00CA")
```

Sets the color to be used for values that are near the specification to purple, as shown below.

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| AC1 | Current | **1.126 mA** | **Nominal a...** | | pass | 1.125 mA | 1.128 mA | **1.128 mA** | **1.125 mA** |
| AC1 | UGF | 546.4 MHz | > 540000000 | | near | 510.8 MHz | 716.5 MHz | 716.5 MHz | 510.8 MHz |
| AC1 | Gain | 46.74 dB | | | | 46.27 dB | 48.42 dB | 48.42 dB | 46.27 dB |
| AC1 | Voffset | 3.065 mV | range -10m... | | pass | 2.349 mV | 3.281 mV | 2.349 mV | 3.281 mV |

## textColorForSpecPassInResults

Specifies a text color, color name or a color value in hexadecimal format, to be used for the *Pass/Fail*, *Min*, and *Max* columns for the result values that meet the specification.

To set this variable in .cdsinit or CIW:

```
maestro.gui textColorForSpecPassInResults "colorName_or_hexadecimalValue"
```


© 2023 Cadence Design Systems, Inc.

| | |
|---|---|
| **Variable Type** | string |
| **Default Value** | "#008000" |
| **Acceptable Values** | Color name or hexadecimal value |

### Example

```
envSetVal("maestro.gui" "textColorForSpecPassInResults" 'string "#00ACAC")
```

Sets the color to be used for values that pass the spec results to light blue, as shown below.

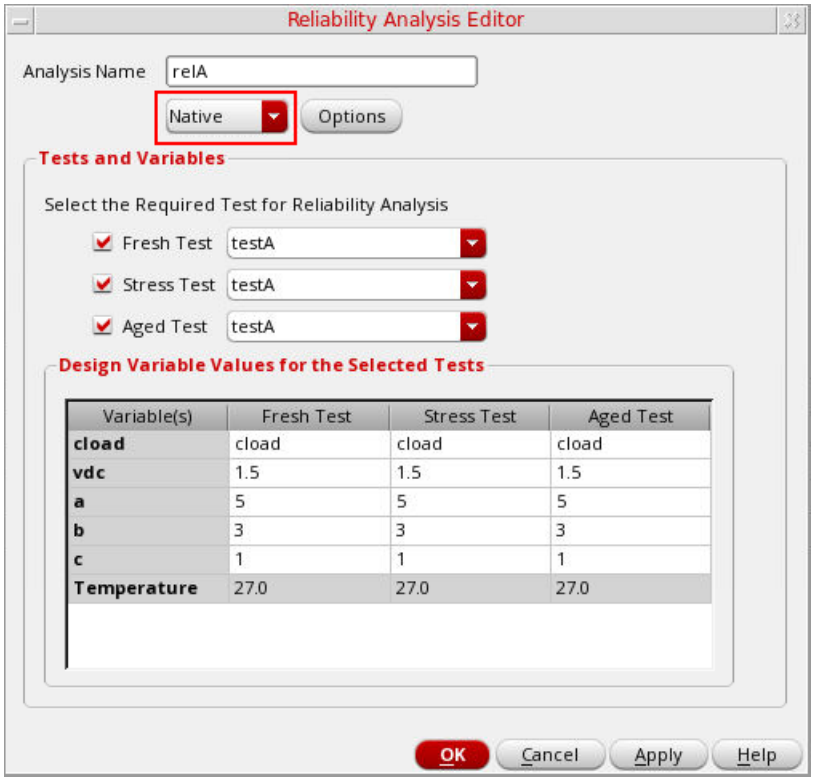
| Point | Test | Output | Nominal | Spec | Weight | Pass/Fail | Min | Max | C0_0 | C0_1 |
|---|---|---|---|---|---|---|---|---|---|---|
| 3 | AC1 | UGF | 546.4 MHz | > 900000000 | | fail | 510.8 MHz | 716.5 MHz | 716.5 MHz | 510.8 MHz |
| 3 | AC1 | Gain | 46.74 dB | | | | 46.27 dB | 48.42 dB | 48.42 dB | 46.27 dB |
| 3 | AC1 | Voffset | 3.065 mV | range -10m... | | pass | 2.349 mV | 3.281 mV | 2.349 mV | 3.281 mV |

## showNativeRelxpertSwitch

Enables you to switch between the Spectre Native and RelXpert modes.


© 2023 Cadence Design Systems, Inc.

**Note:** When this environment variable is set to `t`, a drop-down list is displayed on the *Reliability Analysis Editor* form, as shown in the following figure.




Using this drop-down list, you can choose one of the following simulator modes:

- Native (Spectre Native)
- RelXpert

To set this variable in the .cdsenv:

```
maestro.gui showNativeRelxpertSwitch boolean nil
```

In .cdsinit:

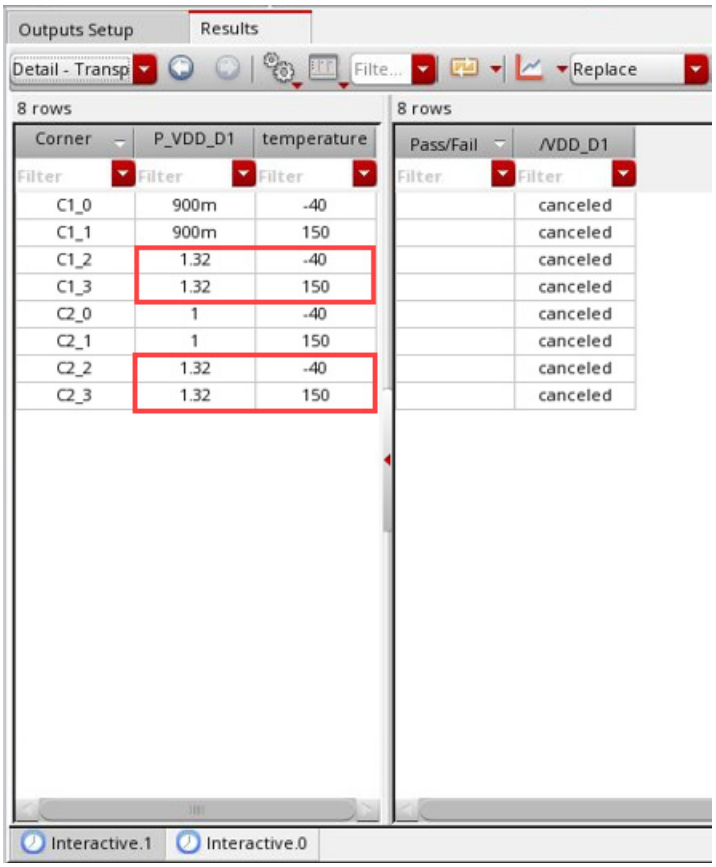
```
envSetVal("maestro.gui" "showNativeRelxpertSwitch" 'boolean nil)
```


© 2023 Cadence Design Systems, Inc.

| | |
|---|---|
| **Variable Type** | boolean |
| **Default Value** | `nil` |
| **Acceptable Values** | ■ `t` — Enables you to switch between the Spectre Native and RelXpert simulator modes using a drop-down list in the Reliability Analysis Editor form.<br>■ `nil` — The drop-down list to switch between the simulator modes is not displayed in the Reliability Analysis Editor form. |

## showRedundantPointsinDT

Displays the redundant data points in the *Detail-Transpose* results view.




To set this variable in the .cdsenv:


© 2023 Cadence Design Systems, Inc.

```
maestro.gui showRedundantPointsinDT boolean nil
```

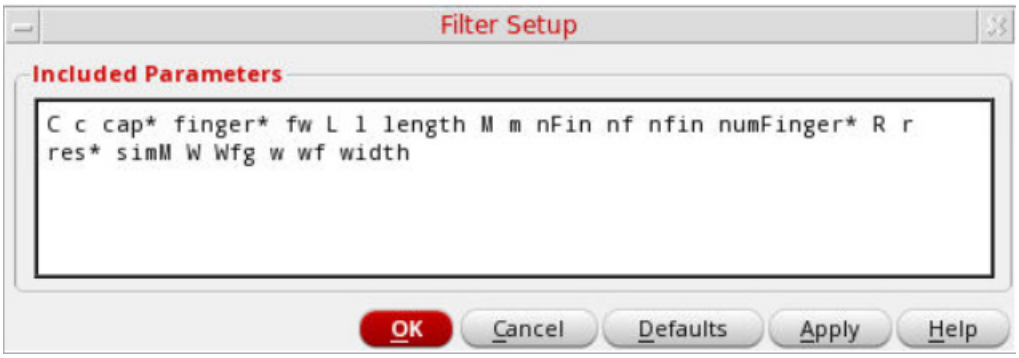
In .cdsinit:

```
envSetVal("maestro.gui" "showRedundantPointsinDT" 'boolean nil)
```

| | |
|---|---|
| **Variable Type** | boolean |
| **Default Value** | `nil` |
| **Acceptable Values** | `t`, `nil` |

## defaultParametersAllowList

Modifies the list of parameters to be displayed in the *Parameters* tab of the *Variables and Parameters* assistant.




To set this variable in the .cdsenv:

```
maestro.gui defaultParametersAllowList string "C c cap* finger* fw L
l length M m nFin nf nfin numFinger* R r res* simM W Wfg w wf width"
```

In .cdsinit:

```
envSetVal("maestro.gui" "defaultParametersAllowList" 'string "C c
cap* finger* fw L l length M m nFin nf nfin numFinger* R r res* simM
W Wfg w wf width")
```


© 2023 Cadence Design Systems, Inc.

| | |
|---|---|
| **Variable Type** | string |
| **Default Value** | `"C c cap* finger* fw L l length M m nFin nf nfin numFinger* R r res* simM W Wfg w wf width"` |
| **Acceptable Values** | The list of parameters to be displayed in the *Parameters* tab of the *Variables and Parameters* assistant. |

## defaultParametersAssistantFilter

Specifies the default filter to be set for device instance parameters in the *Parameters* tab of the *Variables and Parameters* assistant.

To set this variable in the .cdsenv:

```
maestro.gui defaultParametersAssistantFilter string "Allow List"
```

In .cdsinit:

```
envSetVal("maestro.gui" "defaultParametersAssistantFilter" 'string
"Allow List")
```

| | |
|---|---|
| **Variable Type** | string |
| **Default Value** | `"Allow List"` |
| **Acceptable Values** | The default filter to be set for device instance parameters in the *Parameters* tab of the *Variables and Parameters* assistant. |


© 2023 Cadence Design Systems, Inc.

## enableGroupByTagsForVars

Selects the *Group By Tags* option in the context menu for the section *Design Variables* or *Global Variables* in the Data View assistant. It then groups the variables in both the sections according to the assigned tags, creating sub-trees named `tag:`*`tagname`*.

**Note:** Variables that do not have any assigned tags are displayed in a sub-tree named `Tag:untagged`.

To set this variable in the .cdsenv:

```
maestro.gui enableGroupByTagsForVars boolean nil
```

In .cdsinit:

```
envSetVal("maestro.gui" "enableGroupByTagsForVars" 'boolean nil)
```

| | |
|---|---|
| **Variable Type** | Boolean |
| **Default Value** | `nil` |
| **Acceptable Values** | `t`, `nil` |


© 2023 Cadence Design Systems, Inc.

## enableGroupByTagsForCorners

```
maestro.gui enableGroupByTagsForCorners boolean { t | nil }
```

### Description

Selects the *Group By Tags* option in the context menu for the *Corners* section in the Data View assistant. It then groups the corners according to the assigned tags, creating sub-trees named `Tag:`*`tagname`*.

**Note:** Corners that do not have any assigned tags are displayed in a sub-tree named `Tag:untagged`.

The default value is `nil`.

### GUI Equivalent

| | |
|---|---|
| Command: | *Data View Assistant – Corners* |
| Field: | *Group By Tags* |

### Examples

```
envGetVal("maestro.gui" "enableGroupByTagsForCorners")
envSetVal("maestro.gui" "enableGroupByTagsForCorners" 'boolean t)
```

### *Related Topics*

Grouping Corners Using Tags


© 2023 Cadence Design Systems, Inc.

## disableInfoAnalysisOutputWhenUIDisabled

```
maestro.gui disableInfoAnalysisOutputWhenUIDisabled boolean { t | nil }
```

### Description

Controls whether the following circuit information analyses should be saved or not when the corresponding check boxes are disabled in the *Save Circuit Information Analysis* table of the Save options form.

- *primitives*
- *subckts*
- *asserts*

The default value is `nil`.

### GUI Equivalent

| | |
|---|---|
| Command: | *Save Options – Save Circuit Information Analysis* |
| Field: | *Enabled* check box corresponding to the information analyses, *primitives*, *subckts*, and *asserts* |

### Examples

```
envGetVal("maestro.gui" "disableInfoAnalysisOutputWhenUIDisabled")
envSetVal("maestro.gui" "disableInfoAnalysisOutputWhenUIDisabled" 'boolean t)
```


© 2023 Cadence Design Systems, Inc.

## disableHighPerformanceSimulationEnableMS

```
maestro.gui disableHighPerformanceSimulationEnableMS boolean { t | nil }
```

### Description

Controls the availability of the *Enable MS* check box in the High-Performance Simulation Options form, which lets you use the mixed-signal options during simulation when the simulation performance mode is set to APS or Spectre X

- `t` means that the *Enable MS* check box is grayed out.
- `nil` lets you select or deselect the *Enable MS* check box. This is the default.

### GUI Equivalent

None

### Examples

```
envGetVal("maestro.gui" "disableHighPerformanceSimulationEnableMS")
envSetVal("maestro.gui" "disableHighPerformanceSimulationEnableMS" 'boolean t)
```

### *Related Topics*

MS Options


© 2023 Cadence Design Systems, Inc.

## hideStringResultsFailures

```
maestro.gui hideStringResultsFailures boolean { t | nil }
```

### Description

Controls the coloring of results that have specifications written as string values. When specifications are written in string format, result values cannot be compared to get the pass, fail, or near status. Set this variable to `t` to avoid coloring of such results.

Default value is `nil`.

### GUI Equivalent

None

### Examples

```
envGetVal("maestro.gui" "hideStringResultsFailures")
```

```
envSetVal("maestro.gui" "hideStringResultsFailures" 'boolean t)
```

### *Related Topics*

Viewing the Simulation Results


© 2023 Cadence Design Systems, Inc.

## defaultParametersAllowList

```
maestro.gui defaultParametersAllowList string { t | nil }
```

### Description

Controls the coloring of results that have specifications written as string values. When specifications are written in string format, result values cannot be compared to get the pass, fail, or near status. Set this variable to `t` to avoid coloring of such results.

Default value is `nil`.

### GUI Equivalent

None

### Examples

```
envGetVal("maestro.gui" "hideStringResultsFailures")
```

```
envSetVal("maestro.gui" "hideStringResultsFailures" 'boolean t)
```


© 2023 Cadence Design Systems, Inc.

## showRemoveTmpADEDirButton

```
maestro.gui showRemoveTmpADEDirButton boolean { t | nil }
```

### Description

Displays or hides the *Remove .tmpADEDir* button from the *Output Setup* toolbar in ADE Explorer and *Tools* toolbar in ADE Assembler.

The default value is `t`.

### GUI Equivalent

None

### Examples

```
envGetVal("maestro.gui" "showRemoveTmpADEDirButton")
```

```
envSetVal("maestro.gui" "showRemoveTmpADEDirButton" 'boolean nil)
```


© 2023 Cadence Design Systems, Inc.

## showMSOptionsInSpectreXTab

```
maestro.gui showMSOptionsInSpectreXTab boolean { t | nil }
```

### Description

Displays or hides the *MS Options* group box in the High-Performance Simulation Options form for Spectre X.

The default value is `nil`.

### GUI Equivalent

None

### Examples

```
envGetVal("maestro.gui" "showMSOptionsInSpectreXTab")
```

```
envSetVal("maestro.gui" "showMSOptionsInSpectreXTab" 'boolean t)
```


© 2023 Cadence Design Systems, Inc.

## deleteUnusedProbes

```
maestro.gui deleteUnusedProbes boolean { t | nil }
```

### Description

Removes the differentiating colour or the highlights when the previously selected nets in the schematic are reselected.

The default value is `nil`.

### GUI Equivalent

None

### Examples

```
envGetVal("maestro.gui" "deleteUnusedProbes")
```

```
envSetVal("maestro.gui" "deleteUnusedProbes" 'boolean t)
```


© 2023 Cadence Design Systems, Inc.

## disableHighPerformanceSimulationSpectreXOption

```
maestro.gui disableHighPerformanceSimulationSpectreXOption boolean { t | nil }
```

### Description

Removes *Spectre X* from the simulation performance mode options in the High-Performance Simulation Options form and selects the *APS* option.

**Note:** The variable is effective only when you set it in `.cdsinit`.

The default value is `nil`.

### GUI Equivalent

None

### Examples

```
envGetVal("maestro.gui" "disableHighPerformanceSimulationSpectreXOption")
```

```
envSetVal("maestro.gui" "disableHighPerformanceSimulationSpectreXOption" 'boolean
t)
```


© 2023 Cadence Design Systems, Inc.

## donotExpandNameForparam_file

```
maestro.gui donotExpandNameForparam_file boolean { t | nil }
```

### Description

Indicates whether relative path to the dynamic parameter file, which is specified in the Choosing Analyses form for transient analysis, must be resolved to its absolute path..

The default value is `t`.

### GUI Equivalent

None

### Examples

```
envGetVal("maestro.gui" "donotExpandNameForparam_file")
```

```
envSetVal("maestro.gui" "donotExpandNameForparam_file" 'boolean nil)
```


© 2023 Cadence Design Systems, Inc.

## showNoiseSummaryinTableWidget

```
maestro.gui showNoiseSummaryinTableWidget boolean { t | nil }
```

### Description

Specifies whether to enable or disable the Noise Summary window that displays the noise summary report in a tabulated format.

The default value is `nil`.

### GUI Equivalent

None

### Examples

```
envGetVal("maestro.gui" "showNoiseSummaryinTableWidget")
envSetVal("maestro.gui" "showNoiseSummaryinTableWidget" 'boolean nil)
```


© 2023 Cadence Design Systems, Inc.

# maestro.license

### checkoutOrder

Controls the license checkout order to be followed for ADE Explorer and ADE Assembler. This variable has the following valid values:

- `Explorer_Assembler` (default): When set, the ADE Explorer license is searched first. If it is not available, then Assembler license is searched and used.
- `Assembler_Explorer`: When set, the ADE Assembler license is searched first. If it is not available, then ADE Explorer license is searched and used.
- `Explorer`: When set, only ADE Explorer license is searched and used.
- `Assembler`: When set, only ADE Assembler license is searched and used

To set this variable in the .cdsinit file or CIW:

```
envSetVal("maestro.license" "checkoutOrder" 'cyclic
"Explorer_Assembler")
```

| | |
|---|---|
| **Variable Type** | cyclic |
| **Default Value** | `Explorer_Assembler` |
| **Acceptable Values** | `Explorer_Assembler`, `Assembler_Explorer`, `Explorer`, `Assembler` |


© 2023 Cadence Design Systems, Inc.

# maestro.mpt

### createNewStrips

Controls whether to plot new leaves for signals in new strips.

To set this variable in the .cdsenv:

```
maestro.mpt createNewStrips boolean nil
```

In .cdsinit:

```
envSetVal("maestro.mpt" "createNewStrips" 'boolean nil)
```

| | |
|---|---|
| **Variable Type** | Boolean |
| **Default Value** | `nil` |
| **Acceptable Values** | ■ `t` — New leaves for signals are plotted in separate strips.<br>■ `nil` — All leaves for the signals are plotted in a single strip. |

### exactMatchOnly

Controls whether to plot only matching leaves using plotting template.

To set this variable in the .cdsenv:

```
maestro.mpt exactMatchOnly boolean nil
```

In .cdsinit:

```
envSetVal("maestro.mpt" "exactMatchOnly" 'boolean nil)
```


© 2023 Cadence Design Systems, Inc.

| | |
|---|---|
| **Variable Type** | Boolean |
| **Default Value** | `nil` |
| **Acceptable Values** | ■ `t` — Only matching leaves of the signal are plotted using plotting templates.<br>■ `nil` — All leaves for the signals are plotted using plotting templates. |

### useColorBank

Controls whether to plot signals using default color bank.

To set this variable in the .cdsenv:

```
maestro.mpt useColorBank boolean nil
```

In .cdsinit:

```
envSetVal("maestro.mpt" "useColorBank" 'boolean nil)
```

| | |
|---|---|
| **Variable Type** | Boolean |
| **Default Value** | `nil` |
| **Acceptable Values** | ■ `t` — Signals are plotted using default color bank.<br>■ `nil` — All signals are plotted using the signal color in the saved plotting template. |


© 2023 Cadence Design Systems, Inc.

# maestro.plotting

### showViVAGraphLaunchTip

Displays the tip providing information that you can open ViVA XL graph as an assistant or as a standalone window. This tip appears when the ViVA XL graph is opened as an assistant for the first time. Set this variable to `nil` to hide this tip.

To set this variable in the .cdsinit file or CIW:

```
envSetVal("maestro.plotting" "showViVAGraphLaunchTip" 'boolean t)
```

| | |
|---|---|
| **Variable Type** | boolean |
| **Default Value** | t |
| **Acceptable Values** | t, nil |


© 2023 Cadence Design Systems, Inc.

## distributedPlotEnabled

```
maestro.plotting distributedPlotEnabled boolean {t | nil}
```

### Description

Specifies whether to enable distributed plot by default. The default value is `nil`, which indicates that distributed plot is disabled by default.

### GUI Equivalent

None

### Examples

```
envGetVal("maestro.plotting" "distributedPlotEnabled")
envSetVal("maestro.plotting" "distributedPlotEnabled" 'boolean t)
```


© 2023 Cadence Design Systems, Inc.

## distributedPlotEnableCIW

```
maestro.plotting distributedPlotEnableCIW boolean {t | nil}
```

### Description

Specifies whether to enable CIW for distributed plot. The default value is `nil`, which indicates that CIW for distributed plot is disabled by default.

### GUI Equivalent

None

### Examples

```
envGetVal("maestro.plotting" "distributedPlotEnableCIW")
envSetVal("maestro.plotting" "distributedPlotEnableCIW" 'boolean t)
```


© 2023 Cadence Design Systems, Inc.

## distributedPlotJobPolicy

```
maestro.plotting distributedPlotJobPolicy string jobPolicyName
```

### Description

Specifies the default job policy for distributed plot. The default value is `""`, which indicates that Virtuoso uses the job policy that is previously used in the setup.

If the user creates a new maestro view, the default job policy is the local machine.

### GUI Equivalent

None

### Examples

```
envGetVal("maestro.plotting" "distributedPlotJobPolicy")
envSetVal("maestro.plotting" "distributedPlotJobPolicy" 'string "LSCS")
```


© 2023 Cadence Design Systems, Inc.

## distributedPlotLoadCdsInit

```
maestro.plotting distributedPlotLoadCdsInit boolean {t | nil}
```

### Description

Specifies whether to load the `.cdsinit` file before launching the distributed plot. In addition to the `.cdsinit` file, the distributed plot process also tries to load the `.distributedPlotInit` file. The default value is `nil`, which indicates that the `.cdsinit` file is not loaded before launching the distributed plot.

### GUI Equivalent

None

### Examples

```
envGetVal("maestro.plotting" "distributedPlotLoadCdsInit")
envSetVal("maestro.plotting" "distributedPlotLoadCdsInit" 'boolean t)
```


© 2023 Cadence Design Systems, Inc.

## distributedPlotLingerTime

```
maestro.plotting distributedPlotLingerTime boolean {t | nil}
```

### Description

Specifies the time in seconds for which the distributed plot process waits before exiting when there is no active plotting window is visible. The default value is `60` second. It is recommended not to set the linger time to less than `30` second.

### GUI Equivalent

The *Linger Time* field in the *Distributed Plot* section of ADE Assembler Plotting/Printing Options and ADE Explorer Plotting/Printing Options forms

### Examples

```
envGetVal("maestro.plotting" "distributedPlotLingerTime")
envSetVal("maestro.plotting" "distributedPlotLingerTime" 'boolean t)
```


© 2023 Cadence Design Systems, Inc.

## distributedPlotPolicyNameFilter

`maestro.plotting distributedPlotPolicyNameFilter string` *`filterName`*

### Description

Specifies the string by which you want to filter the job policies displayed in the J*ob Policy Name* drop-down list in the Job Policy Selection for Distributed Plot form. The default value is "", which indicates that all job policies defined using the Job Policy Setup form are displayed.

If you set this environment variable to `"UNIX"`, only the job policies that contain UNIX in their names are displayed in the *Job Policy Name* drop-down list in the Job Policy Selection for Distributed Plot form. For example: `UNIX_LBS` and `myJobPolicy_UNIX`.

### GUI Equivalent

None

### Examples

```
envGetVal("maestro.plotting" "distributedPlotPolicyNameFilter")
envSetVal("maestro.plotting" "distributedPlotPolicyNameFilter" 'string "UNIX")
```


© 2023 Cadence Design Systems, Inc.

## distributedPlotEnableNonSupportedPlotWarning

```
maestro.plotting distributedPlotEnableNonSupportedPlotWarning boolean {t | nil}
```

### Description

Specifies whether to issue warning messages for waveforms that cannot be plotted by Distributed Plot service. The default value is `t`, which indicates that warning messages are displayed for waveforms that are not supported by Distributed Plot service.

### GUI Equivalent

None

### Examples

```
envGetVal("maestro.plotting" "distributedPlotEnableNonSupportedPlotWarning")
envSetVal("maestro.plotting" "distributedPlotEnableNonSupportedPlotWarning"
'string "UNIX")
```


© 2023 Cadence Design Systems, Inc.

## quickPlotGlobalWriteDisable

```
maestro.plotting quickPlotGlobalWriteDisable boolean {t | nil}
```

### Description

Specifies whether to turn off generating Quick Plot data. The default value is nil, which indicates that Quick Plot data is generated by default.

If you set this environment variable to `t`, generation of Quick Plot data remains disabled during the entire virtuoso session.

### GUI Equivalent

None

### Examples

```
envGetVal("maestro.plotting" "quickPlotGlobalWriteDisable")
envSetVal("maestro.plotting" "quickPlotGlobalWriteDisable" 'boolean t)
```


© 2023 Cadence Design Systems, Inc.

# maestro.polo

## dspfBindingSummary

```
maestro.polo dspfBindingSummary boolean { t | nil }
```

### Description

Prints the DSPF-to-subcircuit binding summary in the CDS.log file.

The default is `nil`.

### GUI Equivalent

None

### Examples

```
envGetVal("maestro.polo" "dspfBindingSummary")
envSetVal("maestro.polo" "dspfBindingSummary" 'boolean t)
```


© 2023 Cadence Design Systems, Inc.

## dspfBindingToCell

```
maestro.polo dspfBindingToCell cyclic { "bindToAll" | "bindToOne" }
```

### Description

Specifies whether the DSPF files can be bound to multiple subcircuits or one.

- `bindToAll` (the default) specifies that the DSPF files can be bound to multiple subcircuits.
- `bindToOne` specifies that the DSPF files can be bound to one subcircuit each.

### GUI Equivalent

None

### Examples

```
envGetVal("maestro.polo" "dspfBindingToCell")
envSetVal("maestro.polo" "dspfBindingToCell" 'cyclic "bindToOne")
```


© 2023 Cadence Design Systems, Inc.

## mspsSkipMappingForExtractedViews

```
maestro.polo mspsSkipMappingForExtractedViews string { "none" | "all" }
```

### Description

Specifies whether to perform or skip signal mapping for all extracted views. When set to none, signals of all extracted views are mapped to the signals in the schematic. When set to all, mapping is skipped for all extracted views including the views extracted using other Cadence tools, for example, Quantus QRC av_extracted view.

When using the extracted views that are not saved by Virtuoso, set this variable to all to improve simulation performance.

**Note:** This variable is used only when set in `.cdsinit` or `.vdsinit` files.

Default value is `none`.

### GUI Equivalent

None

### Examples

```
envGetVal("maestro.polo" "mspsSkipMappingForExtractedViews")
envSetVal("maestro.polo" "mspsSkipMappingForExtractedViews" 'string "all")
```


© 2023 Cadence Design Systems, Inc.

## netlistSmartViewDSPFOnRecreateNetlist

```
maestro.polo netlistSmartViewDSPFOnRecreateNetlist boolean { t | nil }
```

### Description

Recreates the DSPF file for a Smart View when the *Netlist – Recreate* command is selected.

The default is `t`.

### GUI Equivalent

None

### Examples

```
envGetVal("maestro.polo" "netlistSmartViewDSPFOnRecreateNetlist")
envSetVal("maestro.polo" "netlistSmartViewDSPFOnRecreateNetlist" 'boolean nil)
```


© 2023 Cadence Design Systems, Inc.

## supportDspfMasterOption

```
maestro.polo supportDspfMasterOption boolean { t | nil }
```

### Description

Lets the ADE netlister add `-master` options in the `dspf_include` statement to use the same DSPF file for multiple subcircuits.

The default is `nil`.

### GUI Equivalent

None

### Examples

```
envGetVal("maestro.polo" "supportDspfMasterOption")
envSetVal("maestro.polo" "supportDspfMasterOption" 'boolean t)
```


© 2023 Cadence Design Systems, Inc.

## plotSensitivityDataInSeparateSubwindows

```
maestro.plotting plotSensitivityDataInSeparateSubwindows boolean {t | nil}
```

### Description

Specifies whether to plot the results of sensitivity analysis in separate subwindows.

The default value is `t`, which indicates that the results of sensitivity analysis are plotted in separate subwindows.

If you set this environment variable to `nil`, the results are plotted in the same subwindow.

### GUI Equivalent

None

### Examples

```
envGetVal("maestro.plotting" "plotSensitivityDataInSeparateSubwindows")
envSetVal("maestro.plotting" "plotSensitivityDataInSeparateSubwindows" 'boolean
nil)
```


© 2023 Cadence Design Systems, Inc.

# maestro.simulation

### interactiveE

Specifies whether the ICRPs need to submit points directly to a Spectre plugin. When a large number of sweep points or corners are used, this mode enhances the performance of a simulation run. This variable has the following valid values:

- `t`—Each ICRP creates an initial netlist for the design and submits all the assigned points along with variable changes to a Spectre plugin. The plugin updates the netlist for each point by using the information about the variables. After the simulation for all the points is complete, Spectre plugin returns the results together for those points to the ICRP. This saves the overhead of Spectre startup, license checks, and netlisting separately for each point.
- `nil`—Each ICRP handles simulation for a single point at a time. For each point, it starts a new session with Spectre, checks out a license, creates a netlist, and runs simulation. After the simulation for a point is complete, Spectre returns the results to the ICRP and kills the session. This involves an overhead of repetitive tasks for each point, and impacts the performance when a simulation run involves a large number of sweep points and corners.

**Note:** This variable is applicable only for the Spectre simulator.

To set this variable in the .cdsinit file or CIW:

```
envSetVal("maestro.simulation" "interactiveE" 'boolean t)
```

| | |
|---|---|
| **Variable Type** | boolean |
| **Default Value** | t |
| **Acceptable Values** | t, nil |

### interactiveRTT

Specifies whether the ICRPs need to submit points directly to a Spectre plugin when you are working in the real-time tuning mode. When a large number of sweep points or corners are used, this mode enhances the performance of a simulation run. This variable has the following valid values:

- `t`—Each ICRP creates an initial netlist for the design and submits all the assigned points along with variable changes to a Spectre plugin. The plugin updates the netlist for each point by using the information about the variables. After the simulation for all the points


© 2023 Cadence Design Systems, Inc.

is complete, Spectre plugin returns the results together for those points to the ICRP. This saves the overhead of Spectre startup, license checks, and netlisting separately for each point.

- `nil`—Each ICRP handles simulation for a single point at a time. For each point, it starts a new session with Spectre, checks out a license, creates a netlist, and runs simulation. After the simulation for a point is complete, Spectre returns the results to the ICRP and kills the session. This involves an overhead of repetitive tasks for each point, and impacts the performance when a simulation run involves a large number of sweep points and corners.

**Note:** This variable is applicable only when you use Spectre to perform real-time tuning.

To set this variable in the .cdsinit file or CIW:

```
envSetVal("maestro.simulation" "interactiveRTT" 'boolean t)
```

| | |
|---|---|
| **Variable Type** | boolean |
| **Default Value** | t |
| **Acceptable Values** | t, nil |

### groupRunE

Enables or disables group submission of points to the available ICRP jobs. This variable has the following valid values:

- `t`— Enables group submission of points. ADE Explorer divides the points to be simulated into groups and submits those to the available ICRP jobs at a time. Instead of communicating with ADE Assembler for each point individually, an ICRP runs the simulation for all the points in the group together and returns the results for the group. This enhances the performance of the simulation run.

  **Note:** When set to `t`, the points are grouped at the beginning of the simulation. Once the simulation is in progress, any updates made to the value of the *Max. Jobs* field in the Job Policy Setup form would not be reflected.

- `nil`— Disables group submission of points. ADE Explorer submits a single point to an ICRP at a time.

To set this variable in the .cdsinit file or CIW:


© 2023 Cadence Design Systems, Inc.

```
envSetVal("maestro.simulation" "groupRunE" 'boolean t)
```

| | |
|---|---|
| **Variable Type** | boolean |
| **Default Value** | `t` |
| **Acceptable Values** | `t`, `nil` |

### minGroupSizeSplitAcrossIdleJobs

Specifies the minimum number of pending points (jobs) with an ICRP awaiting to be run. If the number of idle points are more than this count, they are redistributed to other available ICRPs. This helps in enhancing the overall performance of the simulation run when certain ICRPs are running on slow resources. Therefore, you can set this variable to a value greater than 1 to enable reallocation of points.

For example, If a simulation has 100 points and the *Max Jobs* field on the Job Policy Setup form is set to 2, a set of 50 simulation points are allocated to each one of `job1` and `job2`. If the simulations running on `job1` are completed faster than those running on `job2`, `job1` become idle. If you have specified the minimum size for a group of simulations to be considered for the reallocation of points to 5 and the number of points pending with `job2` is greater than 5, some of the points are reallocated to `job1`. This improves the overall run time for simulations.

Note the following:

- This variable is used when groupRunE is set to `t` in ADE Explorer and when groupRunA is set to `t` in ADE Assembler.
- This variable is not used for the Monte Carlo run mode, which uses the `adexl.monte minGroupSizeSplitAcrossIdleJobs`.

To set this variable in the .cdsinit file or CIW:

```
envSetVal( "maestro.simulation" "minGroupSizeSplitAcrossIdleJobs"
'int t)
```

| | |
|---|---|
| **Variable Type** | integer |
| **Default Value** | 0 |


© 2023 Cadence Design Systems, Inc.

| | |
|---|---|
| **Acceptable Values** | ■ `0` or `1` - Reallocation of jobs is disabled.<br>■ Any positive integer value greater than 1 - Reallocation of jobs is enabled when the group size is greater than the specified value. |

### checkSpectreStatus

Enables or disables checking of the status of Spectre processes running simulations. When this check is enabled, ADE Explorer or ADE Assembler periodically checks for the status of the Spectre processes. If a Spectre process is suspended because of the unavailability of resources, ADE Explorer or ADE Assembler also suspends the corresponding job connected to that process.

To set this variable in the .cdsinit file or CIW:

```
envSetVal("maestro.simulation" "checkSpectreStatus" 'boolean t)
```

| | |
|---|---|
| **Variable Type** | boolean |
| **Default Value** | t |
| **Acceptable Values** | t, nil |

### maxRDBSyncWait

Specifies the maximum wait time (in seconds) for which ADE Explorer or ADE Assembler needs to wait for the results data to be used for the evaluation of `calcVal` expressions before reporting an evaluation error. ADE Assembler considers this variable only when the results for the corresponding point are not found in the results database when querying outputs for `calcVal` evaluation. This variable is particularly useful when the simulations are running over NFS and there could be a delay in retrieving the results

The wave comparison feature of ADE Assembler also uses `calcVal` internally. If the wave comparison results are not displayed correctly, you can use this variable to allow some time for data retireval before comparisons are done.

To set this variable in the .cdsenv:

```
maestro.simulation maxRDBSyncWait int 60
```

In .cdsinit or the CIW:


© 2023 Cadence Design Systems, Inc.

```
envSetVal( "maestro.simulation" "maxRDBSyncWait" 'int 60)
```

| | |
|---|---|
| **Variable Type** | Integer |
| **Default Value** | 60 |
| **Acceptable Values** | A positive integer value between the range of `1` and `600` |

### addHostNameToDisplayPort

Specifies whether or not ADE Explorer or ADE Assembler must add a host name if it is missing from the DISPLAY shell environment variable. When you have multiple computer resources available to run simulation, you might skip the host name and provide only the port number, say `:10.0`, in the DISPLAY environment variable. In that case, the tool can use any remote machine to run simulation.

However, this value of DISPLAY may not be considered as complete for certain simulation mode and the tool displays an error. For example, an AMS simulation in interactive mode requires the complete information in the format *`hostName`*:*`portNumber`*. In such a case, if this environment variable is set to t, the tool can add the host name, which is the current host on which Virtuoso is running, to the display port.

To set this variable in the .cdsenv:

```
maestro.simulation addHostNameToDisplayPort boolean t
```

In .cdsinit or the CIW:

```
envSetVal( "maestro.simulation" "addHostNameToDisplayPort" 'boolean
t)
```


© 2023 Cadence Design Systems, Inc.

| | |
|---|---|
| **Variable Type** | boolean |
| **Default Value** | `t` |
| **Acceptable Values** | ■ `t`: If the host name is missing in the value of DISPLAY, the tool adds the host name to the display port<br>■ `nil`: If the host name is missing in the value of DISPLAY, the tool does not add the host name to the display port |

## maestroViewEnvName

`maestro.simulation maestroViewEnvName string` *`variableName`*

### Description

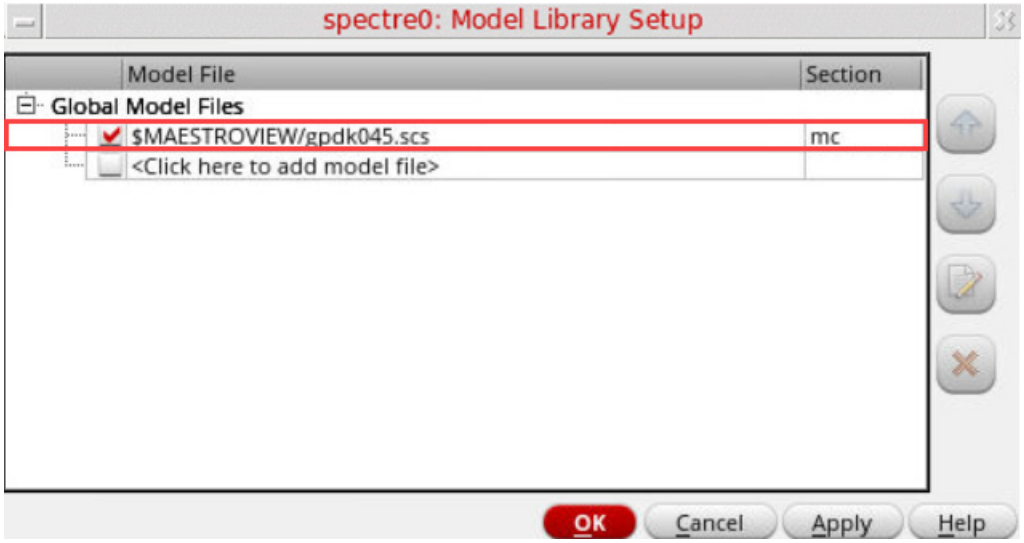
Specifies the name of the shell variable that returns the UNIX path to a maestro cellview.

The default value is `""`.

This shell variable can then be used to specify the path to a model file in the Model Library Setup form.


© 2023 Cadence Design Systems, Inc.

For example, on specifying a model file path as `$MAESTROVIEW/gdpdk045.scs` means that the model file `gdpdk045.scs` is in the directory of the current maestro cellview. The path to this cellview is automatically evaluated through the shell variable `$MAESTROVIEW`.




### GUI Equivalent

None

### Examples

```
envSetVal("maestro.simulation" "maestroViewEnvName")
envSetVal("maestro.simulation" "maestroViewEnvName" 'string "$MAESTROVIEW")
```

## cancelRunAfterRetries

```
maestro.simulation cancelRunAfterRetries int number_of_attempts
```

### Description

Specifies the maximum number of unsuccessful attempts of successful simulation of a point after which you want to cancel the complete simulation run without attempting simulation of the remaining points. The default value is `0`, which implies that ADE Explorer or ADE Assembler continues to retry failing simulations until the run is manually stopped.


© 2023 Cadence Design Systems, Inc.

## GUI Equivalent

None

## Examples

```
envGetVal("maestro.simulation" "cancelRunAfterRetries")
envSetVal("maestro.simulation" "cancelRunAfterRetries" 'int 10)
```


© 2023 Cadence Design Systems, Inc.

# maestro.stimuli

### maxStopTime

Specifies the max value of stop time for any new stimuli defined in the Stimuli Assignment form. It is used to prevent long simulations for waveform previews.

To set this variable in .cdsinit or CIW:

```
envSetVal("maestro.stimuli" "maxStopTime" 'string "1m")
```

| | |
|---|---|
| **Variable Type** | string |
| **Default Value** | "1m" |
| **Acceptable Values** | Any string value |


© 2023 Cadence Design Systems, Inc.

# maestro.stimuliTool

### stopTime

Specifies the stop time for any new stimuli defined in the Stimuli Assignment form.

To set this variable in .cdsinit or CIW:

```
envSetVal("maestro.stimuliTool" "stopTime" 'string "1n")
```

| | |
|---|---|
| **Variable Type** | string |
| **Default Value** | "1n" |
| **Acceptable Values** | Any string value |
| **GUI Equivalent** | Field*, STOP TIME,* in the Stimuli Assignment form |

### isourceResistorValue

Specifies the resistance value of a resistor required to plot waveform from an isource component.

In .cdsinit or the CIW:

```
envSetVal( "maestro.stimuliTool" "isourceResistorValue" 'string
"1k")
```

| | |
|---|---|
| **Variable Type** | string |
| **Default Value** | "1k" |
| **Acceptable Values** | Valid column names, including user-defined column names |


© 2023 Cadence Design Systems, Inc.

# maestro.variables

### autoSyncGlobalVarsToTestVars

Controls whether the value of global and local variables should be in synchronization when modified using ADE Explorer or Assembler.

When this variable is set to `t`, if you are working in ADE Assembler and change a global variable, then every test in ADE Assembler takes that global value to be the design value. If you change a design variable in Explorer and then go to ADE Assembler, then only the local value of design variable for that test is updated. While changing a variable value in ADE Explorer, the value of corresponding global variable is not updated; however, a yellow highlight appears on that global variable indicating that it has been overridden by the test-specific local value.

When this variable is set to `nil`, changing a global variable in ADE Assembler does not affect that variable in the tests. Similarly, changing a design variable in ADE Explorer changes the variable only for that test.

To set this variable in the .cdsinit file or CIW:

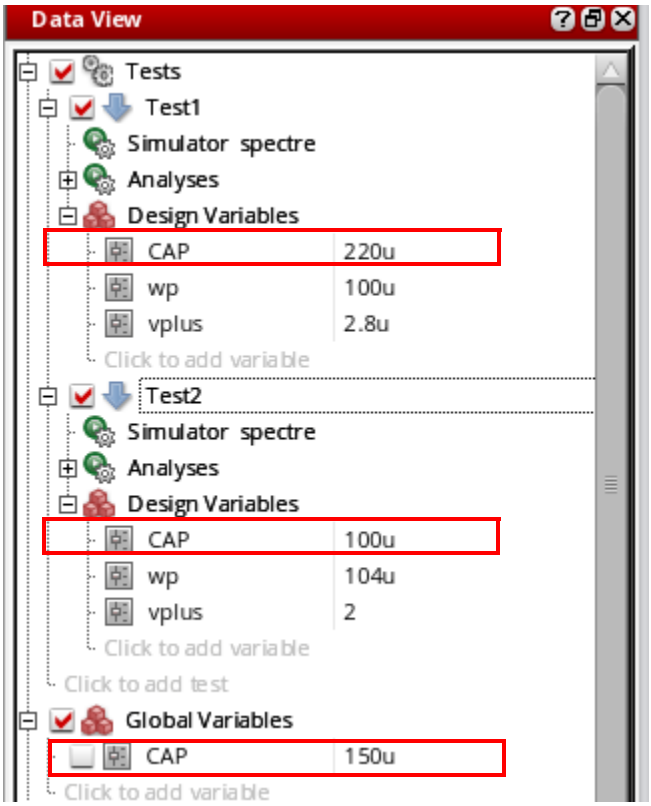
```
envSetVal("maestro.variables" "autoSyncGlobalVarsToTestVars"
'boolean t)
```

| | |
|---|---|
| **Variable Type** | boolean |
| **Default Value** | nil |
| **Acceptable Values** | t, nil |


© 2023 Cadence Design Systems, Inc.

### Example

When you are in ADE Assembler, you can choose to use either global or local design variables. Suppose, you have two tests in ADE Assembler, `Test1` and `Test2` with a common variable, `CAP`. In Test1, `CAP=220u`, and in `Test2`, `CAP=100u`.



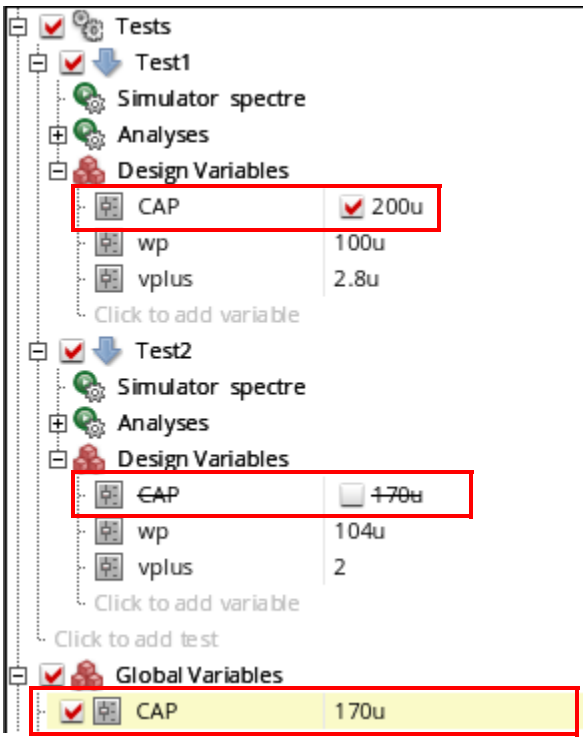

**Case1**: Global variable is not selected. The local values of tests are used in both ADE Assembler and ADE Explorer.

**Case2**: When you select the global variable, `CAP=150u`, the local values are striked-out showing that global value has been used for these tests.

Now, if you open `Test1` in ADE Explorer, the value of `CAP` is `170u`. If you change this value to `200u`, the local value of this variable is changed to `200` in ADE Assembler and a yellow


© 2023 Cadence Design Systems, Inc.

highlight appears on the global value indicating that the global value is overwritten by the local value for a test.





© 2023 Cadence Design Systems, Inc.

# maestro.window

### width

Sets the width (in pixels) of the ADE Explorer and ADE Assembler windows. When the width is 0, the width of the session window is used.

By default, the window is displayed at the pixel coordinates specified by the list `(120:180 1520:980)`. Therefore, the default width of the window is `1400` pixels.

To set this variable in the `.cdsinit` file or CIW:

```
envSetVal("maestro.window" "width" 'int 1200)
```

| | |
|---|---|
| **Variable Type** | integer |
| **Default Value** | `1400` |
| **Acceptable Values** | any integer value |

### height

Sets the height (in pixels) of the ADE Explorer and ADE Assembler windows. When the height is 0, the height of the session window is used.

By default, the window is displayed at the pixel coordinates specified by the list `(120:180 1520:980)`. Therefore, the default height of the window is `800` pixels.

To set this variable in the `.cdsinit` file or CIW:

```
envSetVal("maestro.window" "height" 'int 600)
```

| | |
|---|---|
| **Variable Type** | integer |
| **Default Value** | `800` |
| **Acceptable Values** | any integer value |


© 2023 Cadence Design Systems, Inc.

# Spectre

- analysisOrder
- autoDisplay
- autoDisplayBBox
- controlMode
- currents
- definitionFiles
- dspfFile
- fastViewOption
- includePath
- infoOptions
- modelFiles
- nestlvl
- netlistBBox
- nport_default_interp
- paramRangeCheckFile
- printComments
- resetSpectreApsSharedOptions
- save
- saveahdlvars
- setEngNotation
- simExecName
- simOutputFormat
- stimulusFile
- stopViewList
- subcktprobelvl


© 2023 Cadence Design Systems, Inc.

- uniMode
- useprobes
- finalTimeOp

## AC Match Analysis Environment Variables

- log
- mth
- oprobe
- annotate
- force
- groupby
- oppoint
- prevoppoint
- readns
- restart
- save
- skipdc
- useprevic
- additionalParams
- incrType
- outType
- rangeType
- span
- center
- start
- stop
- step


© 2023 Cadence Design Systems, Inc.

- P
- n
- stepTypeLin
- stepTypeLog


© 2023 Cadence Design Systems, Inc.

# Spectre.emirOpts

- advancedTable
- anaType
- autoLoadVfiFile
- dspfListBox
- dynamicAnaSummaryBox
- editSummaryInfoTable
- emDataFile
- emOnlyIctFile
- enableEMIRAna
- hideAnalysesItems
- ictFile
- includeFile
- layerName
- netName
- qrcTechFile
- short
- shortResLayer
- solverMethod
- solverMethodTable
- spgsAdvIRDrop
- spgsFilterMoscap
- staticAnaSummaryBox
- summaryListBox
- techFileMode
- timeWindowListBox


© 2023 Cadence Design Systems, Inc.

## advancedTable

`spectre.emirOpts advancedTable string` *`any_string`*

### Description

Specifies any additional options that Spectre supports for EMIR analysis.

For more information, refer to the EMIR Analysis section in the *Spectre Classic Simulator, Spectre APS, Spectre X, and Spectre XPS User Guide*.

The default value is `""`.

### GUI Equivalent

| | |
|---|---|
| Command: | *EMIR Analysis Setup – Basic* |
| Field: | Voltus-Fi - *Base Name* |

### Examples

```
envGetVal("spectre.emirOpts" "advancedTable")
envSetVal("spectre.emirOpts" "advancedTable" 'string "((t \"Tlife\" \"100000hours\") (t \"include\" \"settings.conf\"))")
```


© 2023 Cadence Design Systems, Inc.

## anaType

`spectre.emirOpts anaType string` *`Dynamic`*

### Description

Specifies the type of EMIR analysis to be used during simulation.

The valid values are:

- `Dynamic`: Estimates the IR and EM current density by performing a DC or transient simulation.
- `Static`: Enables you to evaluate IR drops and EM currents based on the specified current consumptions for subcircuit instances without running a transient or DC simulation. The specified currents are distributed to the tap devices based on the width and length ratios of devices in the design. The IR drop and EM current analysis is performed based on the current at each tap device.
- `SPGS`: Used to calculate pin-to-tap resistances based on the description of a DSPF file and the options set in an EMIR configuration file. The resistances calculated by SPGS are electrically equivalent resistances, and not the sum of resistances.

The default value is `Dynamic`.

### GUI Equivalent

| | |
|---|---|
| Command | *EMIR Analysis Setup – Basic* |
| Field | Basic - *Type* |

### Examples

```
envGetVal("spectre.emirOpts" "anaType")
envSetVal("spectre.emirOpts" "anaType" 'string "Dynamic")
envSetVal("spectre.emirOpts" "anaType" 'string "Static")
envSetVal("spectre.emirOpts" "anaType" 'string "SPGS")
```


© 2023 Cadence Design Systems, Inc.

## autoLoadVfiFile

```
spectre.emirOpts autoLoadVfiFile boolean { t | nil }
```

### Description

Loads the `Voltus-Fi` setup from the previous EMIR Quantus xDSPF run.

The default value is `nil`.

### GUI Equivalent

| | |
|---|---|
| Command | *EMIR Analysis Setup – Voltus-Fi* |
| Field | Voltus-Fi - *read settings from .vfiExtractDSPFGuiLast* |

### Examples

```
envGetVal("spectre.emirOpts" "autoLoadVfiFile")
envSetVal("spectre.emirOpts" "autoLoadVfiFile" 'boolean nil)
envSetVal("spectre.emirOpts" "autoLoadVfiFile" 'boolean t)
```


© 2023 Cadence Design Systems, Inc.

## dspfListBox

```
spectre.emirOpts dspfListBox string any_string
```

### Description

Specifies the DSPF file to be used during the EMIR analysis. It contain the post-layout data of the design, including fingered devices and net parasitics, with the parasitic and instance sections describing the circuit to be analyzed.

### GUI Equivalent

| | |
|---|---|
| Command | *EMIR Analysis Setup – Techfile* |
| Field | Techfile - *DSPF File* |

### Examples

```
envGetVal("spectre.emirOpts" "dspfListBox")
envSetVal("spectre.emirOpts" "dspfListBox" 'string "((\"$WORKDIR/bias_amp.dspf\" \"\" \"\" \"Not Run\"))")
```


© 2023 Cadence Design Systems, Inc.

## dynamicAnaSummaryBox

`spectre.emirOpts dynamicAnaSummaryBox string` *`any_string`*

### Description

Specifies the type of analysis to be performed and the nets or instances for which the analysis is performed.

It is only effective when anaType is set to `Dynamic`.

### GUI Equivalent

| | |
|---|---|
| Command | *EMIR Analysis Setup – Basic* |
| Field | Basic - *Analysis table* |

### Examples

```
envGetVal("spectre.emirOpts" "dynamicAnaSummaryBox")
envSetVal("spectre.emirOpts" "dynamicAnaSummaryBox" 'string "((t \"instName\" \"\" \"\" \"\" \"\" \"\" \"\" \"\" nil \"\" \"\" \"\" \"\" \"\" \"\" \"\"))")
```


© 2023 Cadence Design Systems, Inc.

## editSummaryInfoTable

```
spectre.emirOpts editSummaryInfoTable boolean { t | nil }
```

### Description

Enables editing the summary information table.

The default value is `nil`.

### GUI Equivalent

| | |
|---|---|
| Command | *EMIR Analysis Setup – Basic* |
| Field | Basic - *Edit Summary Information Table* |

### Examples

```
envGetVal("spectre.emirOpts" "editSummaryInfoTable")
envSetVal("spectre.emirOpts" "editSummaryInfoTable" 'boolean nil)
envSetVal("spectre.emirOpts" "editSummaryInfoTable" 'boolean t)
```


© 2023 Cadence Design Systems, Inc.

## emDataFile

`spectre.emirOpts emDataFile string` *`any_string`*

### Description

Specifies the EM data file to be used during EMIR analysis.

It is only effective when techFileMode is set to `emData File`.

The default value is `" "`.

### GUI Equivalent

| | |
|---|---|
| Command | *EMIR Analysis Setup – Techfile* |
| Field | Techfile - *EM Data File* |

### Examples

```
envGetVal("spectre.emirOpts" "emDataFile")
envSetVal("spectre.emirOpts" "emDataFile" 'string "$WORKDIR/dev_checks.scs")
```


© 2023 Cadence Design Systems, Inc.

## emOnlyIctFile

```
spectre.emirOpts emOnlyIctFile string any_string
```

### Description

Specifies the EM Only ICT file to used during EMIR analysis.

This file provides the process and the EM model information for EM analysis. This is a text-based file, which means you can use any text editor to enter information in this file.

The default value is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *EMIR Analysis Setup – Techfile* |
| Field | Techfile - *EM Only ICT File* |

### Examples

```
envGetVal("spectre.emirOpts" "emOnlyIctFile")
envSetVal("spectre.emirOpts" "emOnlyIctFile" 'string "$WORKDIR/em.ict")
```


© 2023 Cadence Design Systems, Inc.

## enableEMIRAna

```
spectre.emirOpts enableEMIRAna boolean { t | nil }
```

### Description

Enables EMIR analysis during a simulation run.

The default value is `nil`, in which case EMIR analysis is not enabled.

### GUI Equivalent

| | |
|---|---|
| Command | *EMIR Analysis Setup – Basic* |
| Field | Basic - *Enable EMIR Analysis in Transient or DC Simulation* |

### Examples

```
envGetVal("spectre.emirOpts" "enableEMIRAna")
envSetVal("spectre.emirOpts" "enableEMIRAna" 'boolean nil)
envSetVal("spectre.emirOpts" "enableEMIRAna" 'boolean t)
```


© 2023 Cadence Design Systems, Inc.

## hideAnalysesItems

`spectre.emirOpts hideAnalysesItems string` *`any_string`*

### Description

Hides the columns corresponding to the specified analyses from the table displayed when you select the *Dynamic* analysis type in the *Analysis* section of the *Basic* tab.

Note the following:

- The valid value is a space-separated list of analysis names.
- The analysis names must be in lower case.
- There should not be any space in the analysis names.

The default value is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *EMIR Analysis Setup – Basic* |
| Field | Basic - *Analyses - Dynamic* |

### Examples

```
envGetVal("spectre.emirOpts" "hideAnalysesItems")
envGetVal("spectre.emirOpts" "hideAnalysesItems" 'string "irmax iravg emmax
emavg")
```


© 2023 Cadence Design Systems, Inc.

## ictFile

`spectre.emirOpts ictFile string` *`any_string`*

### Description

Specifies the ICT file to be used during EMIR analysis.

This is an editable text file that specifies the EM reliability rules for EM analysis.

The default value is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *EMIR Analysis Setup – Techfile* |
| Field | Techfile - *ICT File* |

### Examples

```
envGetVal("spectre.emirOpts" "ictFile")
envGetVal("spectre.emirOpts" "ictFile" 'string "$WOKRDIR/emData.txt")
```


© 2023 Cadence Design Systems, Inc.

## includeFile

`spectre.emirOpts includeFile string` *`any_string`*

### Description

Specifies any additional files that spectre supports for EMIR analysis.

For more information, refer to the EMIR Analysis section in the *Spectre Classic Simulator, Spectre APS, Spectre X, and Spectre XPS User Guide*.

The default value is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *EMIR Analysis Setup – Techfile* |
| Field | Techfile - *Include File* |

### Examples

```
envGetVal("spectre.emirOpts" "includeFile")
envGetVal("spectre.emirOpts" "includeFile" 'string "WORKDIR/static_ifile.txt")
```


© 2023 Cadence Design Systems, Inc.

## layerName

`spectre.emirOpts layerName string` *`any_string`*

### Description

Specifies the name of the name of the DSPF layer to be shorted.

The default value is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *EMIR Analysis Setup – Basic* |
| Field | Basic - *LayerName* |

### Examples

```
envGetVal("spectre.emirOpts" "layerName")
envGetVal("spectre.emirOpts" "layerName" 'string "Layer1")
```


© 2023 Cadence Design Systems, Inc.

## netName

`spectre.emirOpts netName string` *`any_string`*

### Description

Specifies the name of the net for which the static power grid solver analysis is to be performed.

It is only effective when anaType is set to `SPGS`.

The default value is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *EMIR Analysis Setup – Basic* |
| Field | Basic - *Net Name* |

### Examples

```
envGetVal("spectre.emirOpts" "netName")
envGetVal("spectre.emirOpts" "netName" 'string "net05")
```


© 2023 Cadence Design Systems, Inc.

## qrcTechFile

```
spectre.emirOpts qrcTechFile string any_string
```

### Description

Specifies the QRC techology file to be used during EMIR analysis.

This file contains the current limits for each process layer and is usually available through a foundry.

The default value is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *EMIR Analysis Setup – Techfile* |
| Field | Techfile - *qrcTechFile* |

### Examples

```
envGetVal("spectre.emirOpts" "qrcTechFile")
envGetVal("spectre.emirOpts" "qrcTechFile" 'string "$WORKDIR/em.ict")
```


© 2023 Cadence Design Systems, Inc.

## short

`spectre.emirOpts short string` *`any_string`*

### Description

Specifies the resistors that need to be shorted during static power grid analysis

It is only effective when anaType is set to `SPGS`.

The default value is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *EMIR Analysis Setup – Basic* |
| Field | Basic - *short* |

### Examples

```
envGetVal("spectre.emirOpts" "short")
envSetVal("spectre.emirOpts" "short" 'string "1.5")
```


© 2023 Cadence Design Systems, Inc.

## shortResLayer

`spectre.emirOpts shortResLayer string` *`yes`*

### Description

Confirms if the specified DSPF layer should be shorted or not.

It is only effective when anaType is set to `SPGS`.

The default value is `yes`.

### GUI Equivalent

| | |
|---|---|
| Command | *EMIR Analysis Setup – Basic* |
| Field | Basic - *Short_res_layer* |

### Examples

```
envGetVal("spectre.emirOpts" "shortResLayer")
envSetVal("spectre.emirOpts" "shortResLayer" 'string "yes")
envSetVal("spectre.emirOpts" "shortResLayer" 'string "no")
```


© 2023 Cadence Design Systems, Inc.

## solverMethod

`spectre.emirOpts solverMethod string` *`Direct`*

### Description

Specifies the solver method to be used for EMIR analysis. The valid values are:

- `Direct`: When high accuracy is needed, a brute-force simulation of the entire system—circuit and parasitic resistances and capacitances—can be performed to accurately calculate EMIR for any net. This approach is also called the one-step method, where the EMIR simulation performance and capacity are subject to the limitations of the circuit simulator being used.
- `Iterated`: To conduct EMIR simulation on circuits with much larger power and signal nets, within a shorter time, this alternative approach that involves decoupling the nonlinear circuit simulation from the linear RC net analysis. You can iterate the linear RC net analysis by modifying the layout. However, the nonlinear circuit simulation is done only once. This approach is also called the two-step method.

The default value is `Direct`.

### GUI Equivalent

| | |
|---|---|
| Command | *EMIR Analysis Setup – Solver* |
| Field | Solver - *Solver Method* |

### Examples

```
envSetVal("spectre.emirOpts" "solverMethod")
envSetVal("spectre.emirOpts" "solverMethod" 'string "Direct")
envSetVal("spectre.emirOpts" "solverMethod" 'string "Iterated")
```


© 2023 Cadence Design Systems, Inc.

## solverMethodTable

```
spectre.emirOpts solverMethodTable string any_string
```

### Description

Specifies the solver method options that spectre supports with EMIR analysis.

The default value is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *EMIR Analysis Setup – Solver* |
| Field | Solver - *Solver Method table* |

### Examples

```
envSetVal("spectre.emirOpts" "solverMethodTable")
envSetVal("spectre.emirOpts" "solverMethodTable" 'string "((t \"solver\"
\"solverOptions\"))")
```


© 2023 Cadence Design Systems, Inc.

## spgsAdvIRDrop

```
spectre.emirOpts spgsAdvIRDrop boolean { t | nil }
```

### Description

Enables or disables the advanced IR drop analysis.

It is only effective when anaType is set to `SPGS`.

The default value is `nil`.

### GUI Equivalent

| | |
|---|---|
| Command | *EMIR Analysis Setup – Basic* |
| Field | Basic - *Advanced IR Drop Analysis* |

### Examples

```
envGetVal("spectre.emirOpts" "spgsAdvIRDrop")
envSetVal("spectre.emirOpts" "spgsAdvIRDrop" 'boolean t)
envSetVal("spectre.emirOpts" "spgsAdvIRDrop" 'boolean nil)
```


© 2023 Cadence Design Systems, Inc.

## spgsFilterMoscap

`spectre.emirOpts spgsFilterMoscap string` *`no`*

### Description

Specifies if the tap nodes connected to MOSFET devices should filtered out or not.

It is only effective when anaType is set to `SPGS`.

The default value is "`no`".

### GUI Equivalent

| | |
|---|---|
| Command | *EMIR Analysis Setup – Basic* |
| Field | Basic - *filter moscap* |

### Examples

```
envGetVal("spectre.emirOpts" "spgsFilterMoscap")
envSetVal("spectre.emirOpts" "spgsFilterMoscap" 'string "no")
envSetVal("spectre.emirOpts" "spgsFilterMoscap" 'string "yes")
```


© 2023 Cadence Design Systems, Inc.

## staticAnaSummaryBox

`spectre.emirOpts staticAnaSummaryBox string` *`any_string`*

### Description

Specifies the type of analysis to be performed and the nets or instances for which the analysis is performed.

It is effective only when anaType is set to `Static`.

The default value is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *EMIR Analysis Setup – Basic* |
| Field | Basic - *Analysis table* |

### Examples

```
envGetVal("spectre.emirOpts" "staticAnaSummaryBox")
```

```
envSetVal("spectre.emirOpts" "staticAnaSummaryBox" 'string "((t \"I0.*\" t t t
\"additionalOptions\") (t \"<Click_To_Add>\" nil nil \"\")"))
```


© 2023 Cadence Design Systems, Inc.

## summaryListBox

`spectre.emirOpts summaryListBox string` *`any_string`*

### Description

Specifies the summary information in the summary information table.

It is only effective when editSummaryInfoTable is set to `t`.

The default value is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *EMIR Analysis Setup – Summary* |
| Field | Summary - *Summary Information* |

### Examples

```
envSetVal("spectre.emirOpts" "summaryListBox")
```

```
envSetVal("spectre.emirOpts" "summaryListBox" 'string "((t
\"techfile=$WORKDIR/em.ict\" t t t \"value\") (t \"<Click_To_Add>\" nil nil
\"\")"))
```


© 2023 Cadence Design Systems, Inc.

## techFileMode

`spectre.emirOpts techFileMode string` *`qrcTechFile with external EM Only ICT File`*

### Description

Specifies the mode to set up the EM technology files. The valid values are:

- `qrcTechFile with external EM only ICT File`: Specifies a QRC technology and an EM only ICT file.
- `qrcTechFile with embedded emModel section`: Specifies a QRC technology file.
- `ICT File with external EM only ICT File`: Specifies an ICT file and an EM only ICT file.
- `ICT File with embedded emModel section`: Specifies an ICT file.
- `ICT File with embedded emModel section`: Specifies an EM data file.

The default value is `qrcTechFile with external EM Only ICT File`.

### GUI Equivalent

| | |
|---|---|
| Command | *EMIR Analysis Setup – Techfile* |
| Field | Techfile - *Mode* |

### Examples

```
envGetVal("spectre.emirOpts" "techFileMode")
envSetVal("spectre.emirOpts" "techFileMode" 'string "qrcTechFile with external EM Only ICT File")
envSetVal("spectre.emirOpts" "techFileMode" 'string "qrcTechFile with embedded emModel section")
envSetVal("spectre.emirOpts" "techFileMode" 'string "ICT File with external EM Only ICT File")
envSetVal("spectre.emirOpts" "techFileMode" 'string "ICT File with embedded emModel section")
envSetVal("spectre.emirOpts" "techFileMode" 'string "emData File")
```


© 2023 Cadence Design Systems, Inc.

## timeWindowListBox

`spectre.emirOpts timeWindowListBox string` *`any_string`*

### Description

Specifies the time period for which the EMIR analysis is to be run.

The default value is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *EMIR Analysis Setup – Solver* |
| Field | Solver - *Time Window table* |

### Examples

```
envGetVal("spectre.emirOpts" "timeWindowListBox")
```

```
envSetVal("spectre.emirOpts" "timeWindowListBox" 'string "((\"0\" \"100n\"))")
```


© 2023 Cadence Design Systems, Inc.

# Spectre.envlpOpts

## sweepmethod

```
spectre.envlpOpts sweepmethod string { coarse | fine | userdefined | adaptive }
```

### Description

Controls the sweep algorithm during fast envelope circuit characterization.

- `coarse`: Uses seven magnitude points and twelve phase points when the parameter `fastmethod` is set to `baseband`.
- `fine`: Uses ten magnitude points and sixteen phase points when the parameter `fastmethod` is set to `baseband`.
- `userdefined`: Lets you define the grid using the parameter `sweepnum`.
- `adaptive`: Automatically determines the number of sweep points.

The default value is `coarse`.

### GUI Equivalent

| | |
|---|---|
| Command | *Envelope Following Options—Accuracy* |
| Field | *sweepmethod* |

### Examples

```
envGetVal("spectre.envlpOpts" "sweepmethod")
```

```
envSetVal("spectre.envlpOpts" "sweepmethid" 'string "fine")
```


© 2023 Cadence Design Systems, Inc.

# Spectre.envOpts

## case_sensitive

Enables case sensitivity in a regenerated netlist and adds `case_sensitive = true` in the `dspf_include` statement for a Smart View.

For more information, see DSPF Support.

To set this variable in the .cdsinit file or CIW, use the call:
```
envSetVal("spectre.envOpts" "case_sensitive" 'string "true")
```

To set this variable in the .cdsenv file, use the call:
```
spectre.envOpts case_sensitive 'string "true"
```

| | |
|---|---|
| **Variable Type** | string |
| **Default Value** | `true` |
| **Acceptable Values** | `true`, `false` |
| **GUI Equivalent** | N/A |

## lsusp

Selects or deselects the check box *Check in License When Suspended* in the Environment Options form for Spectre.

To set this variable in the .cdsinit file or CIW, use the call:
```
envSetVal("spectre.envOpts" "lsusp" 'boolean nil)
```

To set this variable in the .cdsenv file, use the call:
```
spectre.envOpts lsusp boolean nil
```

| | |
|---|---|
| **Variable Type** | boolean |
| **Default Value** | `nil` |
| **Acceptable Values** | `t`, `nil` |
| **GUI Equivalent** | *Environment Options form — Check in License When Suspended* |


© 2023 Cadence Design Systems, Inc.

## lsuspEnvOverridesState

Overrides the setting specified through lsusp.

**Note:** In case you want to set this variable in the CIW, ensure that you do so before opening a maestro view.

To set this variable in the .cdsinit file or CIW, use the call:
```
envSetVal("spectre.envOpts" "lsuspEnvOverridesState" 'boolean nil)
```

To set this variable in the .cdsenv file, use the call:
```
spectre.envOpts lsuspEnvOverridesState boolean nil
```

| | |
|---|---|
| **Variable Type** | boolean |
| **Default Value** | `nil` |
| **Acceptable Values** | `t`, `nil` |
| **GUI Equivalent** | N/A |


© 2023 Cadence Design Systems, Inc.

## setTopLevelAsSubckt

```
spectre.envOpts setTopLevelAsSubckt boolean { t | nil }
```

### Description

Controls whether the top-level schematic should be netlisted as a subcircuit or not.

This variable is supported with the si batch mode flow and must be set in the .simrc file.

The default is `nil`.

### GUI Equivalent

None

### Examples

```
envGetVal("spectre.envOpts" "setTopLevelAsSubckt")
```

```
envSetVal("spectre.envOpts" "setTopLevelAsSubckt" 'boolean t)
```


© 2023 Cadence Design Systems, Inc.

# Spectre.hb

## dynchoice

```
spectre.hb dynchoice string { Parameter vector| Parameter file }
```

### Description

Specifies the type of input dynamic parameter.

- `Parameter vector`: Lets you specify the time and value for the specified parameter.
- `Parameter file`: Lets you specify a parameter file.

The default value is `Parameter vector`.

### GUI Equivalent

| | |
|---|---|
| Command | *Analyses—Choose—Choosing Analyses form* |
| Field | *Parameter* |

### Examples

```
envGetVal("spectre.hb" "dynchoice")
```

```
envSetVal("spectre.hb" "dynchoice" 'string "Parameter file")
```


© 2023 Cadence Design Systems, Inc.

# Spectre.opts

## enable_dcsweep_op_info

```
spectre.opts enable_dcsweep_op_info string { "no" | "yes" }
```

### Description

Specifies whether the *Evaluate over DC Sweep* check box in the Operating Region Specification form is selected or deselected by default.

The default is "no".

### GUI Equivalent

| | |
|---|---|
| Command | *Operating Region assistant—Show All* |
| Field | *Evaluate over DC Sweep* |

### Examples

```
envGetVal("spectre.opts" "enable_dcsweep_op_info")
envSetVal("spectre.opts" "enable_dcsweep_op_info" 'string "yes")
```


© 2023 Cadence Design Systems, Inc.

## tranFilterExtreme

```
spectre.opts tranFilterExtreme boolean { t | nil }
```

### Description

Selects or deselects the option *extreme* for the *Filter* field in the *Transient* tab of the Device Checking Options form.

**Note:** This variable works only for newly created ADE Explorer or ADE Assembler views.

The default is `nil`.

### GUI Equivalent

| | |
|---|---|
| Command | *Device Checking Options form —Transient tab — Filter* |
| Field | *extreme* |

### Examples

```
envGetVal("spectre.opts" "tranFilterExtreme")
envSetVal("spectre.opts" "tranFilterExtreme" 'boolean t)
```

### *Related Topics*

Specifying the Global Device Check Options


© 2023 Cadence Design Systems, Inc.

# Spectre.pss

The following table lists the environment variables in the spectre.pss partition. These variables work exactly same as the variables in the spectre.hb partition.

| spectre.pss environment variable | Corresponding spectre.hb environment variable |
|---|---|
| dynchoice | dynchoice |


© 2023 Cadence Design Systems, Inc.

# Spectre.tran

- dynamicChoice
- dynamicParameter
- dynParamChoice
- dynParamInput
- dynParamSetChoice
- dynParamSetInput

## dynamicChoice

Specifies the dynamic parameter choice.

If set to *Vector*, the time value pairs specified in the table are used during simulation. On setting this variable to *file*, the parameter value is taken from the selected file.

To set this variable in the `.cdsinit` file or CIW, use the call:
```
envSetVal("spectre.tran" "dynamicChoice" 'string "Vector")
```

To set it in the .cdsenv file, add:
```
spectre.tran dynamicChoice 'string "Vector"
```

| | |
|---|---|
| **Variable Type** | string |
| **Default Value** | `"Vector"` |
| **Acceptable Values** | `Vector, file` |
| **GUI Equivalent** | *Choosing Analyses – Dynamic Parameter – Parameter* or *Parameter Set* |

## dynamicParameter

Specifies the data type of the dynamic parameter.

To set this variable in the `.cdsinit` file or CIW, use the call:
```
envSetVal("spectre.tran" "dynamicParameter" 'string "Parameter")
```


© 2023 Cadence Design Systems, Inc.

To set it in the .cdsenv file, add:
`spectre.tran dynamicParameter 'string "Parameter"`

| | |
|---|---|
| **Variable Type** | string |
| **Default Value** | `"Parameter"` |
| **Acceptable Values** | `Parameter, Parameter Set` |
| **GUI Equivalent** | *Choosing Analyses – Dynamic Parameter* |

## dynParamChoice

Specifies the transient analysis parameter to be varied with time, during a transient simulation.

If set to `User-defined`, you can manually add a parameter name.

**Note:** This variable can be used when the value for the variable `dynamicParameter` is set to `Parameter`.

To set this variable in the `.cdsinit` file or CIW, use the call:
`envSetVal("spectre.tran" "dynParamChoice" 'string "User-defined")`

To set it in the .cdsenv file, add:
`spectre.tran dynParamChoice 'string "User-defined"`

| | |
|---|---|
| **Variable Type** | string |
| **Default Value** | `"User-defined"` |
| **Acceptable Values** | `User-defined`, `errpreset`, `method`, `relref`, `maxstep`, `isnoisy`, `iteratio`, `reltol`, `residu-altol`, `vabstol`, `iabstol`, `temp` and `tnom` |
| **GUI Equivalent** | *Choosing Analyses – Dynamic Parameter – Parameter – Param* |

## dynParamInput

When the variable, `dynParamChoice` is set to `User-defined`, this variable can be used to manually specify the name of the parameter to be varied.


© 2023 Cadence Design Systems, Inc.

**Note:** This variable can be used when the value for the variable `dynamicParameter` is set to `Parameter`.

To set this variable in the `.cdsinit` file or CIW, use the call:
`envSetVal("spectre.tran" "dynParamInput" 'string "")`

To set it in the .cdsenv file, add:
`spectre.tran dynParamInput 'string ""`

| | |
|---|---|
| **Variable Type** | string |
| **Default Value** | `""` |
| **Acceptable Values** | A user-defined string. |
| **GUI Equivalent** | *Choosing Analyses – Dynamic Parameter – Parameter – Param* |

## dynParamSetChoice

Specifies the transient analysis parameter to be varied with time, during a transient simulation.

If set to `User-defined`, you can manually add a parameter name.

**Note:** This variable can be used when the value for the variable `dynamicParameter` is set to `Parameter Set`.

To set this variable in the `.cdsinit` file or CIW, use the call:
`envSetVal("spectre.tran" "dynaramSetChoice" 'string "User-defined")`

To set it in the .cdsenv file, add:
`spectre.tran dynParamSetChoice 'string "User-defined"`

| | |
|---|---|
| **Variable Type** | string |
| **Default Value** | `"User-defined"` |
| **Acceptable Values** | `User-defined`, `errpreset`, `method`, `relref`, `maxstep`, `isnoisy`, `iteratio`, `reltol`, `residu-altol`, `vabstol`, `iabstol`, `temp` and `tnom` |
| **GUI Equivalent** | *Choosing Analyses – Dynamic Parameter – Parameter Set– Param* |


© 2023 Cadence Design Systems, Inc.

## dynParamSetInput

When the variable, `dynParamChoice` is set to `User-defined`, this variable can be used to manually specify the name of the parameter to be varied.

**Note:** This variable can be used when the value for the variable `dynamicParameter` is set to `Parameter Set`.

To set this variable in the `.cdsinit` file or CIW, use the call:
`envSetVal("spectre.tran" "dynParamSetInput" 'string "")`

To set it in the .cdsenv file, add:
`spectre.tran dynParamSetInput 'string ""`

| | |
|---|---|
| **Variable Type** | string |
| **Default Value** | `""` |
| **Acceptable Values** | A user-defined string. |
| **GUI Equivalent** | *Choosing Analyses – Dynamic Parameter – Parameter Set – Param* |


© 2023 Cadence Design Systems, Inc.

# Spectre X


- spectreXAdvanced
- spectreXPreset
- spectreXMpOption
- spectreXEnablePostLayout
- spectreXAccuracyScale
- spectreXHosts
- spectreXMpThreads
- spectreXMtOption
- spectreXNumThreads
- spectreXMTPOption
- uniMode


## spectreXAdvanced

Displays or hides the advanced options, *Preset Override* and *Parasitics Optimization* in the High-Performance Simulation Options form.

To set this variable in the .cdsinit file or CIW, use the call:

```
envSetVal("spectre.turboOpts" "spectreXAdvanced" 'boolean nil)
```

| | |
|---|---|
| **Variable Type** | boolean |
| **Default Value** | nil |
| **Acceptable Values** | ■ `t` –Displays the advanced options on the High-Performance Simulation Options form<br>■ `nil` –Hides the advanced options on the High-Performance Simulation Options form |
| **GUI Equivalent** | *High-Performance Simulation Options–Advanced Options* |


© 2023 Cadence Design Systems, Inc.

## spectreXPreset

Specifies the value of *Preset* for Spectre X.

To set this variable in the .cdsinit file or CIW, use the call:

```
envSetVal("spectre.turboOpts" "spectreXPreset" 'string "MX")
```

| | |
|---|---|
| **Variable Type** | string |
| **Default Value** | "MX" |
| **Acceptable Values** | ■ CX – When a golden simulation reference is needed<br>■ AX – For high-precision analog applications<br>■ MX – For most analog applications (default)<br>■ LX – For power management and other relaxed analog applications<br>■ VX – For custom IC verification |
| **GUI Equivalent** | *High-Performance Simulation Options–Preset* |

## spectreXMpOption

Defines the type of multi-processing. Spectre X allows distributed simulation, therefore, it can used more computer cores across multiple computer hosts to further speed up DC and transient run time of large to very large postlayout circuits. You can select the following methods for job distribution: Farm, SSH, and RSH.

To set this variable in the .cdsinit file or CIW, use the call:

```
envSetVal("spectre.turboOpts" "spectreXMpOption" 'string "SSH")
```

| | |
|---|---|
| **Variable Type** | string |
| **Default Value** | "SSH" |
| **Acceptable Values** | Farm, SSH, RSH |
| **GUI Equivalent** | *High-Performance Simulation Options–Multi-Processing* |


© 2023 Cadence Design Systems, Inc.

## spectreXEnablePostLayout

Enables or disables the option to select parasitics optimization with the chosen preset option for the Spectre X simulation performance mode.

To set this variable in the .cdsinit file or CIW, use the call:

```
envSetVal("spectre.turboOpts" "spectreXEnablePostLayout" 'boolean nil)
```

| | |
|---|---|
| **Variable Type** | boolean |
| **Default Value** | `nil` |
| **Acceptable Values** | ■ `t` – Enables the parasitics optimization on the High-Performance Simulation Options form<br>■ `nil` – Disables the parasitics optimization on the High-Performance Simulation Options form |
| **GUI Equivalent** | *High-Performance Simulation Options– Parasitics Optimization* |

## spectreXAccuracyScale

Specifies the value of Parasitics Optimization.

To set this variable in the .cdsinit file or CIW, use the call:

```
envSetVal("spectre.turboOpts" "spectreXAccuracyScale" 'string "MX")
```

| | |
|---|---|
| **Variable Type** | string |
| **Default Value** | "`MX`" |


© 2023 Cadence Design Systems, Inc.

| | |
|---|---|
| **Acceptable Values** | ■ `OFF` – When parasitics optimization is switched off<br>■ `CX` – When a golden simulation reference is needed<br>■ `AX` – For high-precision analog applications<br>■ `MX` – For most analog applications (default)<br>■ `LX` – For power management and other relaxed analog applications<br>■ `VX` – For custom IC verification |
| **GUI Equivalent** | *High-Performance Simulation Options– Parasitics Optimization* |

## spectreXHosts

Specifies the machine name of the host. For example, `sj-<username>` or `noi-<username>`.

To set this variable in the .cdsinit file or CIW, use the call:

```
envSetVal("spectre.turboOpts" "spectreXHosts" 'string "<machine_name>")
```

| | |
|---|---|
| **Variable Type** | string |
| **Default Value** | `""` |
| **Acceptable Values** | Name of the host to run Spectre X |
| **GUI Equivalent** | N/A |

## spectreXMpThreads

Specifies the number of threads when Multi-Processing is selected in Spectre X simulation performance mode.

To set this variable in the .cdsinit file or CIW, use the call:

```
envSetVal("spectre.turboOpts" "spectreXMpThreads" 'string "")
```


© 2023 Cadence Design Systems, Inc.

| | |
|---|---|
| **Variable Type** | string |
| **Default Value** | `""` |
| **Acceptable Values** | Integer values passed as strings |
| **GUI Equivalent** | *High-Performance Simulation Options–Multi-Processing* |

## spectreXMtOption

Specifies the type of multi-threading in Spectre X simulation performance mode.

To set this variable in the .cdsinit file or CIW, use the call:

```
envSetVal("spectre.turboOpts" "spectreXMtOption" 'string "Auto")
```

| | |
|---|---|
| **Variable Type** | string |
| **Default Value** | "`Auto`" |
| **Acceptable Values** | ■ `Auto` – To automatically allocate the threads for simulation<br>■ `Disable` – To disable multi-threading<br>■ `Manual` – To manually allocate threads |
| **GUI Equivalent** | *High-Performance Simulation Options–Multi-Threading* |

## spectreXNumThreads

Specifies the number of threads for Multi-Threading in Spectre X simulation performance mode.

To set this variable in the .cdsinit file or CIW, use the call:

```
envSetVal("spectre.turboOpts" "spectreXNumThreads" 'string "")
```

| | |
|---|---|
| **Variable Type** | string |


© 2023 Cadence Design Systems, Inc.

| | |
|---|---|
| **Default Value** | `""` |
| **Acceptable Values** | Integer values passed as strings |
| **GUI Equivalent** | *High-Performance Simulation Options–# Threads* |

## spectreXMTPOption

Specifies whether the type of Spectre X is multi-threading or multi-processing.

To set this variable in the .cdsinit file or CIW, use the call:

```
envSetVal("spectre.turboOpts" "spectreXMTPOption" 'string "Multi-Threading")
```

| | |
|---|---|
| **Variable Type** | string |
| **Default Value** | "`Multi-Threading`" |
| **Acceptable Values** | "`Multi-Threading`", "`Multi-Processing`" |
| **GUI Equivalent** | N/A |

## uniMode

This variable is used to set the simulation performance modes for Spectre. The available modes are Spectre (default), APS, Spectre X, and Spectre FX.

To set this variable in the .cdsinit file or CIW, use the call:

```
envSetVal("spectre.turboOpts" "uniMode" 'string  "Spectre")
```

| | |
|---|---|
| **Variable Type** | string |
| **Default Value** | `Spectre` |
| **Acceptable Values** | `Spectre`, `APS`, `Spectre X`, `Spectre FX` |
| **GUI Equivalent** | *Setting up Explorer Environment*– Setting up High-Performance Simulation Options |


© 2023 Cadence Design Systems, Inc.

# spectreVerilog

The following table lists the environment variables in the spectreVerilog partition. These variables work exactly same as the variables in the spectre partition.

| spectreVerilog environment variable | Corresponding spectre environment variable |
|---|---|
| autoLoadVfiFile | autoLoadVfiFile |
| dspfListBox | dspfListBox |
| editSummaryInfoTable | editSummaryInfoTable |
| nportbbsversion | nportbbsversion |
| preserveSubcktTermNamesByOrder | preserveSubcktTermNamesByOrder |


© 2023 Cadence Design Systems, Inc.

# Ultrasim

- fastViewOption
- wf_format


© 2023 Cadence Design Systems, Inc.

# voltus_fi.results

## vfiEnableGUI

```
voltus_fi.results vfiEnableGUI boolean { t | nil }
```

### Description

Specifies whether to hide or display the warning message about the deprecation of Voltus™-Fi Custom Power Integrity Solution in the next IC base release. This message appears when you open the EMIR Analysis Setup form.

The default value is `nil`.

### GUI Equivalent

None

### Examples

```
envGetVal("voltus_fi.results" "vfiEnableGUI")
```

```
envSetVal("voltus_fi.results" "vfiEnableGUI" 'boolean t)
```


© 2023 Cadence Design Systems, Inc.

# Real Time Tuning Environment Variables

- activeTimeout
- timeoutShowLog
- hostMode
- host
- useCustomizedSimDir
- simulateOn
- showSimModeChangeDialog
- keepVariableDependency
- sliderOnMouseOver
- saveToMode

### activeTimeout

Specifies the timeout in seconds for the RTT simulation. If the RTT assistant is inactive for longer than the specified timeout value, the simulator license gets checked in and the RTT simulation is stopped.

To set this variable in the .cdsinit file or CIW:

```
envSetVal("maestro.rtt" "activeTimeout" 'int 3600)
```

| | |
|---|---|
| **Variable Type** | integer |
| **Default Value** | 3600 |
| **Acceptable Values** | Any integer value greater than 0 |
| **GUI Equivalent** | N/A |

### timeoutShowLog

Specifies the timeout in seconds. If the RTT simulation does not complete by the specified time, a pop up message is displayed to show the latest status.

To set this variable in the .cdsinit file or CIW:


© 2023 Cadence Design Systems, Inc.

```
envSetVal("maestro.rtt" "timeoutShowLog" 'int 10)
```

| | |
|---|---|
| **Variable Type** | integer |
| **Default Value** | 10 |
| **Acceptable Values** | Any integer value |
| **GUI Equivalent** | N/A |

### hostMode

Specifies the host as local or remote on which you want to run the RTT simulation.

To set this variable in the .cdsinit file or CIW:

```
envSetVal("maestro.rtt" "hostMode" 'string "local")
```

| | |
|---|---|
| **Variable Type** | string |
| **Default Value** | local |
| **Acceptable Values** | remote, local |
| **GUI Equivalent** | N/A |

### host

If the host mode is set to `remote`, specify the host name in this variable that can be used to run the remote simulation.

To set this variable in the .cdsinit file or CIW:

```
envSetVal("maestro.rtt" "host" 'string "")
```

| | |
|---|---|
| **Variable Type** | string |
| **Default Value** | NULL |
| **Acceptable Values** | Any string value |


© 2023 Cadence Design Systems, Inc.

### useCustomizedSimDir

Specifies the RTT simulation directory if you do not want to use the default directory, `/dev/shm`, or this default directory is not ready. When the value of this variable is NULL, that means the shared memory is used.

To set this variable in the .cdsinit file or CIW:

```
envSetVal("maestro.rtt" "useCustomizedSimDir" 'string "")
```

| | |
|---|---|
| **Variable Type** | string |
| **Default Value** | NULL |
| **Acceptable Values** | Any string value |

### simulateOn

Specifies an event used to control the time lag that occurs between the change of values for variables and parameter adjustment and simulation run in RTT. By default, this variable is set to `onMouseUp` event, which starts the simulation when mouse button is released on slider.

To set this variable in the .cdsinit file or CIW:

```
envSetVal("maestro.rtt" "simulateOn" 'cyclic "onMouseUp")
```

| | |
|---|---|
| **Variable Type** | string |
| **Default Value** | onMouseUp |
| **Acceptable Values** | onMouseUp, adaptiveDelay, and 500ms |

### showSimModeChangeDialog

When simulations in RTT cannot be run in interactive mode, displays a dialog in which you can choose whether you want to launch RTT or not. When this variable is set to `nil`, RTT is always launched irrespective of the interactive or batch mode.

To set this variable in the .cdsinit file or CIW:


© 2023 Cadence Design Systems, Inc.

```
envSetVal("maestro.rtt" "showSimModeChangeDialog" 'boolean t)
```

| | |
|---|---|
| **Variable Type** | boolean |
| **Default Value** | t |
| **Acceptable Values** | t, nil |

### keepVariableDependency

Maintains the dependency of variable or parameters in the RTT mode, if it exists. When this variable is set to `nil`, the dependent values can also be tuned.

To set this variable in the .cdsinit file or CIW:

```
envSetVal("maestro.rtt" "keepVariableDependency" 'boolean t)
```

| | |
|---|---|
| **Variable Type** | boolean |
| **Default Value** | t |
| **Acceptable Values** | t, nil |

### sliderOnMouseOver

Makes the multiple sliders visible in the RTT assistant. By default, this variable is set to `t`, which means slider appears only when you mouse-hover or click a variable or parameter. You can set this variable to `nil` to show the sliders for all the variables and parameters.

To set this variable in the .cdsinit file or CIW:

```
envSetVal("maestro.rtt" "sliderOnMouseOver" 'boolean t)
```

| | |
|---|---|
| **Variable Type** | boolean |
| **Default Value** | t |
| **Acceptable Values** | t, nil |


© 2023 Cadence Design Systems, Inc.

### saveToMode

Controls the default option to backannotate variables and parameters from the RTT assistant to ADE Explorer and schematic from the *Save back* drop-down menu.

To set this variable in the `.cdsinit` file or CIW, use the following call:

```
envSetVal("maestro.rtt" "saveToMode" 'cyclic "explorer")
```

| | |
|---|---|
| **Variable Type** | cyclic |
| **Default Value** | `explorer` |
| **Acceptable Values** | ■ `explorer`: Updates the design variable and parameter values in the setup assistant.<br>■ `schematic`: Updates the parameter values in the schematic.<br>■ `explorerschematic` : Updates the values of the design variables and parameters in the setup assistant, and of the parameters in the schematic. |
| **GUI Equivalent** | *Launch – Plugins – Real Time Tuning – Save back* |


© 2023 Cadence Design Systems, Inc.

# Environment Variables


- showViVAGraphLaunchTip
- showSaveBySubcktAssistant
- dimSameColor
- height
- dimSameColor
- showWhatsNewWindow
- outputSetupFilteringNumOutputThreshold
- interactiveE
- interactiveRTT
- groupRunE
- minGroupSizeSplitAcrossIdleJobs
- autoSyncGlobalVarsToTestVars
- autoSyncGlobalVarsToTestVars
- innerSweepSwappableSizeLimit
- allowSignalsExpressionInSameSubwindow
- waveformReaderThreadCount
- useQPDataToCreateDataSheet
- headerAlignmentSide
- headerAlignmentSide
- headerTruncationDirection
- headerTruncationWidth
- detailtransposeViewShowDefault
- maestro.variables
- nameDisplayWidthInDataView
- showFilterInDataView



© 2023 Cadence Design Systems, Inc.

- checkSpectreStatus
- confirmBeforeTestOrCornerDelete
- whatToSaveDefault
- usePlottingTemplate
- Spectre
- keepHistoryData
- spectreMTNumberUsed
- CheckForNewCellviewVarsUponRun
- keepHistoryData
- alwaysNameTerminalOutputs
- terminalSelectionType
- stopTime
- maxStopTime
- defaultOutputColumns
- isourceResistorValue
- maxRDBSyncWait
- exportFormat
- displayImplicitSignals
- saveImplicitSignalsDuringRun
- hideShowMultiProcess
- showUnitsInResults
- textColorForSpecFailInResults
- textColorForSpecNearInResults
- textColorForSpecPassInResults


© 2023 Cadence Design Systems, Inc.

### dimSameColor

Sets the color of traces in waveform info-balloons to match with the color of same traces in the ViVA Graph assistant.

Controls the color of waveforms that are previously plotted in waveform info-balloon and ViVA Graph assistant. When this variable is set to `t`, the color of existing waveforms becomes dimmer every time a new simulation run takes place. When this variable is set to `nil`, the waveforms may appear in different colors as defined in the ViVA color bank.

**Note:** This variable works only for the RTT simulations.

To set this variable in the .cdsinit file or CIW:

```
envSetVal("viva.graph" "dimSameColor" 'string "true")
```

| | |
|---|---|
| **Variable Type** | string |
| **Default Value** | true |
| **Acceptable Values** | true, false |

### dimOldTraces

Sets the color of existing traces to dim whenever a new trace is appended to the ViVA Graph window. The trace with brightest color is the most-recent trace in the graph window. When this variable is set to `true`, which is the default value, the colors of already existing traces in the graph are dimmer as compared to the newly-appended traces. When this variable is set to `false`, all the existing and new traces are displayed in the same color.

**Note:** This variable works only for the RTT simulations.

To set this variable in the .cdsinit file or CIW:

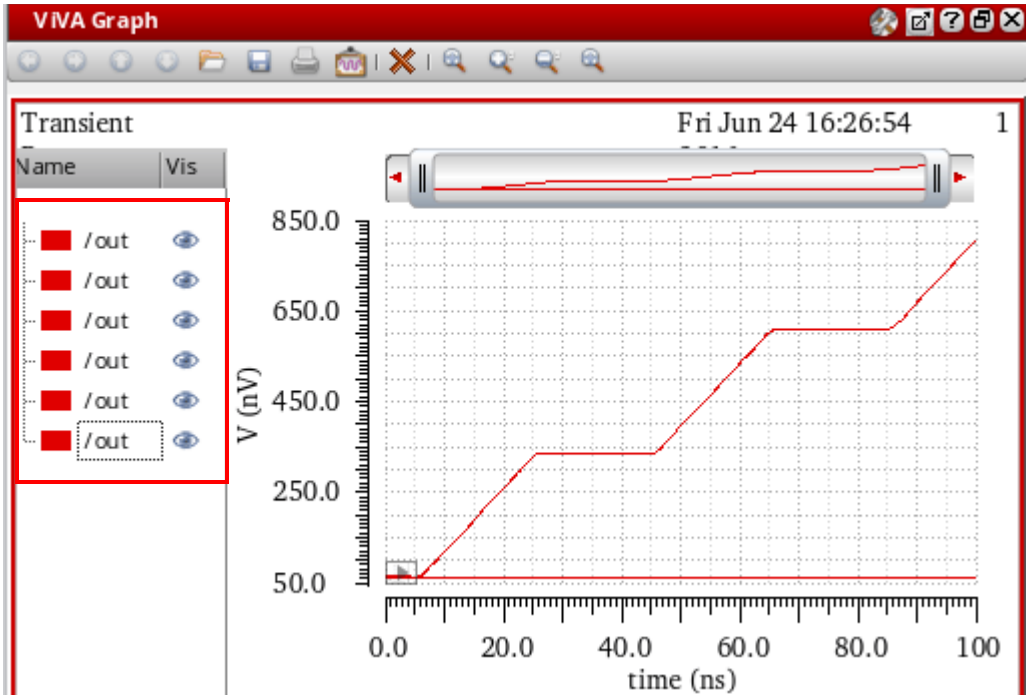
```
envSetVal("viva.graph" "dimOldTraces" 'string "false")
```

| | |
|---|---|
| **Variable Type** | string |
| **Default Value** | true |
| **Acceptable Values** | true, false |


© 2023 Cadence Design Systems, Inc.

### *Example*

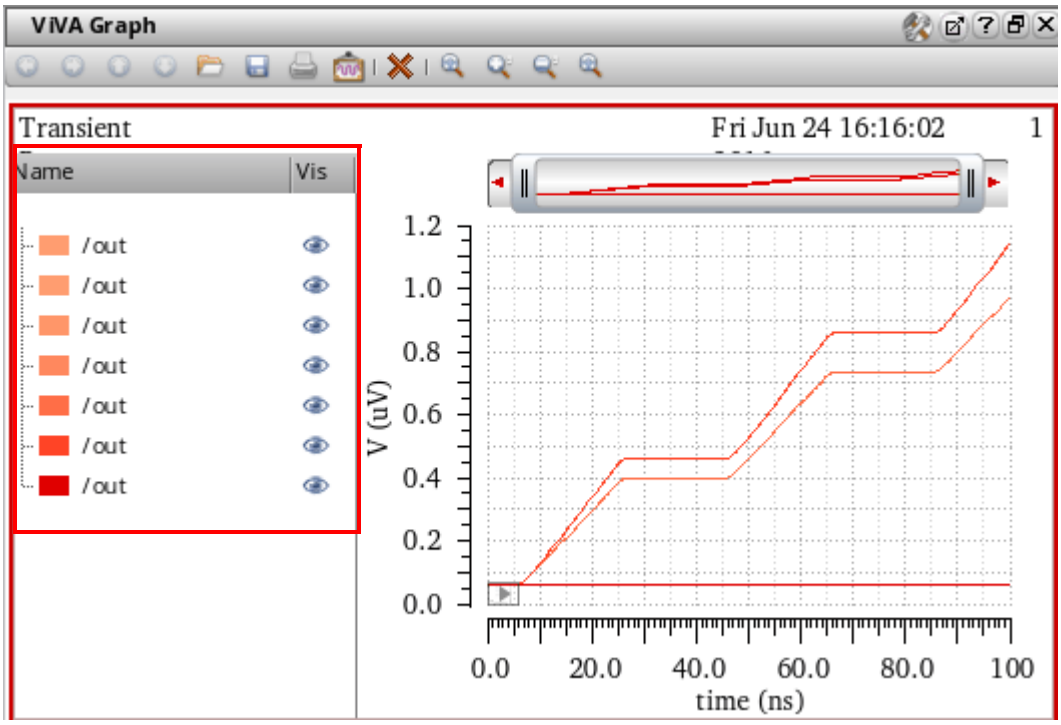
When the variable is set to `false`:





© 2023 Cadence Design Systems, Inc.

When the variable is set to `true`:




### innerSweepSwappableSizeLimit

Specifies the limit for the number of simulation points up to which the `swapSweep` function can be successfully run.

**Note:** The `swapSweep` function swaps the X-axis value with the specified sweep variable. If the number of points are more than the limit specified by this variable, you need to input a particular X-axis point to be used for swapping.

To set this variable in the .cdsinit file or CIW:

```
envSetVal("asimenv.plotting" "innerSweepSwappableSizeLimit" 'int
500)
```

| | |
|---|---|
| **Variable Type** | integer |
| **Default Value** | 500 |
| **Acceptable Values** | Any integer value greater than 0 |


© 2023 Cadence Design Systems, Inc.

### allowSignalsExpressionInSameSubwindow

Plots signals and expressions in the same subwindow. This variable works when you select some expressions and/or signals from the results, right-click them, and choose *Plot*, *Plot All,* or *quickPlotAll* options. When this variable is set to `t` (default value), all signals and expression waveforms are plotted in a same subwindow. When this variable is set to `nil`, signals and expression waveforms are plotted in different subwindows.

To set this variable in the .cdsinit file or CIW:

```
envSetVal("asimenv" "allowSignalsExpressionInSameSubwindow" 'boolean
t)
```

| | |
|---|---|
| **Variable Type** | boolean |
| **Default Value** | t |
| **Acceptable Values** | t, nil |

### waveformReaderThreadCount

Specifies the number of threads that are computed by SRR while reading a parametric waveform data. The purpose of this variable is to achieve maximum throughput while loading a parametric waveform.

To set this variable in the .cdsinit file or CIW:

```
envSetVal("asimenv.plotting" "waveformReaderThreadCount" 'string
"auto")
```

| | |
|---|---|
| **Variable Type** | string |
| **Default Value** | auto |
| **Acceptable Values** | Any integer value |

The default value, `auto`, indicates that number of threads are computed automatically. When you specify an integer value, it directs the SRR to start the specified number of threads. When you specify `0`, it disables the SRR threading while reading the parametric data.


© 2023 Cadence Design Systems, Inc.

### useQPDataToCreateDataSheet

Determines whether the quick plot data is to be used for printing waveforms in the datasheet. If this variable is set to `t`, the quick plot data is used for printing waveforms in the datasheet. In case the quick plot data does not exist and this variable is set to `t`, the quick plot data will be generated first.

If this variable is set to `nil`, the full waveform data is used and the *Use Quickplot data* option is disabled on the Create Datasheet form. The default value for this variable is `t`, which means quick plot data is used by default for printing waveforms in the datasheet.

To set this variable in the .cdsinit file or CIW:

```
envSetVal("asimenv.plotting" "useQPDataToCreateDataSheet" 'boolean
t)
```

| | |
|---|---|
| **Variable Type** | boolean |
| **Default Value** | t |
| **Acceptable Values** | t, nil |

### headerAlignmentSide

Controls the text alignment of the specification column headers in the *Detail-Transpose* view

To set this variable in the .cdsinit file or CIW:

```
envSetVal("adexl.gui" "headerAlignmentSide" 'string "Center")
```

| | |
|---|---|
| **Variable Type** | string |
| **Default Value** | Center |
| **Acceptable Values** | Left, Right, Center |

### headerTruncationDirection

Specifies the direction along which the column header needs to be truncated. ADE Assembler, ADE Explorer, and ADE XL truncates the column header along the direction specified by this variable when this variable is used with the `headerTruncationWidth` variable, thus retaining the number of characters specified by `headerTruncationWidth`.


© 2023 Cadence Design Systems, Inc.

**Note:** This variable also considers ellipses (...) while truncating the column header.

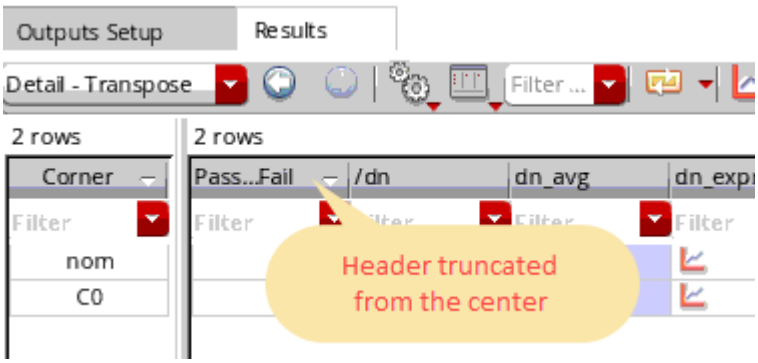
To set this variable in the .cdsinit file or CIW:

```
envSetVal("adexl.gui" "headerTruncationDirection" 'string "Left")
```

| | |
|---|---|
| **Variable Type** | string |
| **Default Value** | Left |
| **Acceptable Values** | Left, Center, Right |

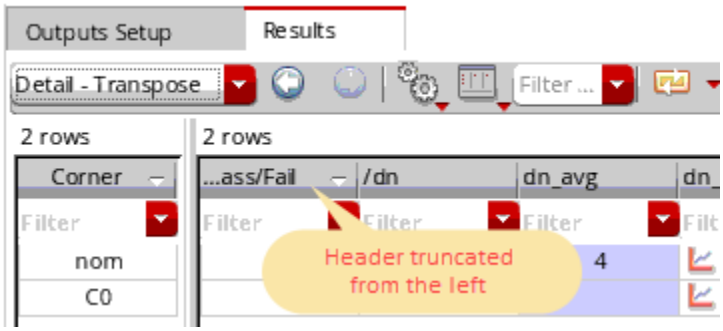
**Left**

Truncates the column header from the left and leaves the characters from the right. For example, if `headerTruncationWidth` is set to 8 and `headerTruncationDirection` is set to Left, the column headers appear as shown below.




**Center**

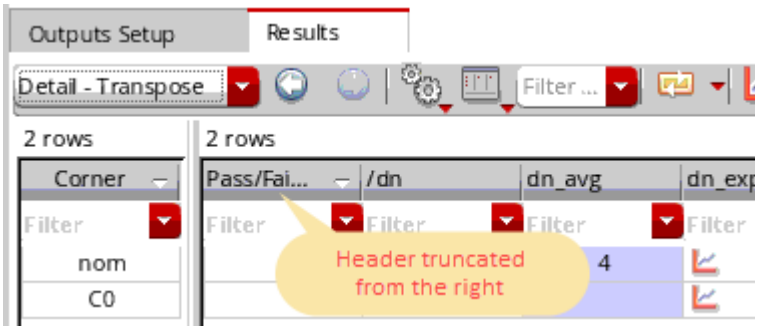
Truncates the column header from the middle and leaves the characters from left and right, as shown in the example below





© 2023 Cadence Design Systems, Inc.

| | |
|---|---|
| **Right** | Truncates the column header from the right and leaves the characters from the left, as shown in the example below. |




### headerTruncationWidth

Specifies character length of the test name that appears in the specification column headers in the *Detail-Transpose* view.

To set this variable in the .cdsinit file or CIW:

```
envSetVal("adexl.gui" "headerTruncationWidth" 'int 30)
```

| | |
|---|---|
| **Variable Type** | integer |
| **Default Value** | 24 |
| **Acceptable Values** | Any positive integer |

### detailtransposeViewShowDefault

Specifies the default columns to be displayed in the *Detail-Transpose* results view.

To set this variable in the .cdsinit file or CIW:

```
envSetVal("adexl.gui" "detailtransposeViewShowDefault" 'string
"\"Scalar Expressions\" \"Waveform Expressions\" \"Signals\"
\"Corner Expressions\" \"Sweep Expressions\" \"Top Level
Expressions\" \"Device Checks\" \"Test Name\" \"Fixed Parameters\"")
```

| | |
|---|---|
| **Variable Type** | String |


© 2023 Cadence Design Systems, Inc.

| | |
|---|---|
| **Default Value** | "\"Scalar Expressions\" \"Waveform Expressions\" \"Signals\" \"Corner Expressions\" \"Sweep Expressions\" \"Top Level Expressions\" \"Any Expressions\" \"Device Checks\"" |
| **Acceptable Values** | A space-separated list of one or more of the following types of columns:<br>■ "\"Scalar Expressions\": Shows or hides the columns that show the scalar results<br>■ \"Waveform Expressions\": Shows or hides the columns that show the waveform results<br>■ \"Signals\": Shows or hides the columns that show the scalar or waveform results of signals in the design<br>■ \"Corner Expressions\": Shows or hides the columns that display the results of outputs measured across corners<br>■ \"Sweep Expressions\": Shows or hides the columns that display the results of outputs measured across sweep points<br>■ \"Top Level Expressions\": Shows or hides the columns that display the results of outputs measured across all dimensions<br>■ \"Any Expressions\": Shows or hides the columns that display the results of outputs measured across any dimension<br>■ \"Device Checks": Shows or hides the columns that display the results of device checks<br>■ \"Test Name\": Shows or hides the test name in the column header of each output<br>■ \"Fixed Parameters\": Shows or hides the columns for the fixed parameters or variables in the left pane of the Detail-Transpose view |

### onHistoryNameCollision

Determines how ADE Explorer saves a history in case another history with the same name


© 2023 Cadence Design Systems, Inc.

already exists. The default history names are `ExplorerRun.0` for edit mode and `ExplorerRORun.0.RO` for read-only mode.

| | |
|---|---|
| **Variable Type** | string |
| **Default Value** | `"Overwrite"` |
| **Acceptable Values** | ■ `"Error"`: ADE Explorer displays an error when the history name for the current simulation run already exists<br>■ `"Overwrite"`: ADE Explorer overwrites the history files, `ExplorerRun.0` and `ExplorerRORun.0.RO`, for every new simulation run<br>■ `"IncrementAsNew"`: ADE Explorer increments the history names for consecutive simulation runs. For example, `ExplorerRun.1`, `ExplorerRun.2`, `ExplorerRORun.1.RO`, and `ExplorerRORun.2.RO` |

### *Important Points to Note*

- If you have set the `adexl.historyNamePrefix showNameHistoryForm` variable to `t` to display the Specify History Name form where you can specify a custom history name, ADE Explorer uses that name instead of `ExplorerRun.0` or `ExplorerRORun.0.RO`. However, the rule to overwrite, increment, or show error in case of a history name conflict is decided based on the value of `onHistoryNameCollision`.
- Related blog post: Virtuosity: Exploring Histories

### CheckForNewCellviewVarsUponRun

Controls the check for new design variables before running simulations in the schematic hierarchy.

To set this variable in the .cdsenv:

```
adexl.test checkForNewCellviewVarsUponRun cyclic "Full"
```

In .cdsinit or the CIW:


© 2023 Cadence Design Systems, Inc.

```
envSetVal( "adexl.test" "checkForNewCellviewVarsUponRun" 'cyclic
"Full")
```

| | |
|---|---|
| **Variable Type** | cyclic |
| **Default Value** | `Full` |
| **Acceptable Values** | ■ `"Full"` – Checks for all new design variables before running simulations. This prevents simulation errors by ensuring that the simulation is run with variables that have valid values.<br>■ `"No"` – Defers the check for new variables. This improves the simulation performance when the simulation hierarchy contains large number of instances.<br>**Note:** When `checkForNewCellviewVarsUponRun` is set to `No`, new variables added in the hierarchy since the last simulation are not identified. This can result in simulation failure.<br>■ `"SimInfoParameters"` – Checks for only those variables that are defined in the CDF `simInfo` section and are related to netlisting or simulation. |

### alwaysNameTerminalOutputs

Specifies whether to always use the _I suffix in the names of outputs that save current data.

To set this variable in .cdsinit or CIW:

```
envSetVal("auCore.selection" "alwaysNameTerminalOutputs" 'boolean
nil)
```


© 2023 Cadence Design Systems, Inc.

| | |
|---|---|
| **Variable Type** | boolean |
| **Default Value** | nil |
| **Acceptable Values** | ■ `t` — Adds the _I suffix to all current type of signal outputs<br>■ `nil` — Does not add the _I suffix |

### terminalSelectionType

Specifies the type of data–current, voltage, or both–to be saved or plotted for a terminal selected for an output.

To set this variable in .cdsinit or CIW:

```
envSetVal("auCore.selection" "terminalSelectionType" 'cyclic
"current")
```

| | |
|---|---|
| **Variable Type** | cyclic |
| **Default Value** | "current" |
| **Acceptable Values** | ■ "current" — Saves only current values for the selected terminals<br>■ "voltage" — Saves only voltage values for the selected terminals<br>■ "both" — Saves both current and voltage values for the selected terminals<br>■ "prompt" — Prompts you to select the type of output to be saved for the selected terminals |
| **GUI Equivalent** | *Terminal Selection Type* drop-down list on the ADE Explorer Plotting/Printing Options form |

Related Topic:

Saving Output Signals


© 2023 Cadence Design Systems, Inc.

### exportFormat

Specifies the format to save the results exported from ADE Explorer or ADE Assembler. You can choose to export the result values without units, values with units in separate columns, or values with units in the same column as displayed on the *Results* tab.

To set this variable in the .cdsenv:

```
adexl.results exportFormat cyclic "numbersOnly"
```

In .cdsinit or the CIW:

```
envSetVal( "adexl.results" "exportFormat" 'cyclic "numbersOnly")
```

| | |
|---|---|
| **Variable Type** | cyclic |
| **Default Value** | "numbersOnly" |
| **Acceptable Values** | ■ "numbersOnly" — Exports the results without units. For example, if a result value for UGF is displayed in the Results table as 8.829 MHz, its unit is not saved in the exported Excel file, as shown below. |

| Test | Output | Nominal | Spec |
|---|---|---|---|
| AMP_SIM | UGF | 8.829e6 | > 7 |
| AMP_SIM | BW 3dB | 7.622e3 | > 8K |

■ "valuesAsInGUI" — Exports the results with units in the same column, as shown below.

| Test | Output | Nominal |
|---|---|---|
| AMP_SIM | UGF | 8.829 MHz |
| AMP_SIM | BW 3dB | 7.622K |

■ "numberAndUnit" — Exports the result values and units in separate columns, as shown below.

| Test | Output | Nominal | unit |
|---|---|---|---|
| AMP_SIM | UGF | 8.829e6 | Hz |
| AMP_SIM | BW 3dB | 7.622e3 | |


© 2023 Cadence Design Systems, Inc.

# C

# Environment Variables for Spectre Simulator Options Form

This section describes the environment variables that you can set in the `.cdsenv` file to specify default values for Spectre simulator options or analysis options.

***Important points to note:***

- The default values specified for the variables in this document indicate the default values specified for these variables in the $CDSHOME/tools/dfII/etc/tools/spectre/.cdsenv file.
- If the default value of a variable is specified as an empty string value(`""`) or as `"default"`, Spectre uses the in-built default value for that variable.

  For example, the default value for `spectre.hb noiseout` is `""`. If you do not modify this value in Virtuoso, while running the simulation, Spectre internally sets it to the in-built default value `"usb"`.

  For more details, refer *Spectre Circuit Simulator Reference Guide*.

You can set or customize the Spectre simulation options using the *Simulator Options* form in ADE. This chapter provides details of all available options on the following tabs of the *Simulator Options* form and their respective environment variables:

- Simulator Options - Main Tab
- Simulator Options - Algorithm Tab
- Simulator Options - Component Tab
- Simulator Options - Check Tab
- Simulator Options - Annotation Tab
- Simulator Options - Miscellaneous Tab


© 2023 Cadence Design Systems, Inc.

# spectre.opts

## Simulator Options - Main Tab

- generalnoiseinst
- generalnoiseinstonoff
- highvoltage
- iabstol
- multithread
- noiseOffType
- noiseOnType
- nthreads
- reltol
- residualtol
- temp
- tempeffects
- tnom
- vabstol

## Simulator Options - Algorithm Tab

- convdbg
- dc_pivot_check
- dptran_gmethod
- gmethod
- gmin
- gmin_check
- gmindc
- homotopy
- icpriority
- ilimit
- nonconv_topnum
- pivabs
- pivotdc
- pivrel
- preorder
- rabsclamp
- rabsshort
- rebuild_matrix
- rforce
- rthresh
- try_fast_op

## Simulator Options - Component Tab

- approx
- auto_minductor
- vth_vdsmin
- ivthl
- ivthn
- ivthp
- ivthw
- imacromodels
- maxrsd
- nport_default_passivity


© 2023 Cadence Design Systems, Inc.

- nportcompress
- nportcompressfiledir
- nportirfiledir
- nportirreuse
- nportunusedportgmin
- nportunusedportrmin
- nportbbsversion
- minr
- scale
- scalefactor
- scalem
- tmevthmod
- vthmod
-

## Simulator Options - Check Tab

- checklimitdest
- checklimitfile
- checklimitskipfile
- checklimitskipsubs
- diagnose
- iccapcheck
- gnshorts
- redefinedparams
- opptcheck
- topcheck

## Simulator Options - Annotation Tab

- audit
- cols
- colslog
- debug
- digits
- error
- info
- inventory
- maxnotes
- maxnotestologfile
- maxwarns
- maxwarnstologfile
- notation
- note
- narrate
- printstep
- simstat
- title
- warn

## Simulator Options - Miscellaneous Tab

- additionalArgs
- ahdllint
- ahdllint_maxwarn
- ahdllint_on
- flow
- quantities


© 2023 Cadence Design Systems, Inc.

- sensbinparam
- sensfileonly
- senstype
- sensfile
- sensformat
- value1


© 2023 Cadence Design Systems, Inc.

# Simulator Options - Main

## checklimitskipinsts

```
spectre.opts checklimitskipinsts string any_string_value
```

### Description

Specifies the array of instances to be skipped in device checking. Instances can have any wildcard symbols.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Check* |
| Field | *checklimitskipinsts* |

## generalnoiseinst

```
spectre.opts generalnoiseinst string any_string_value
```

### Description

List of instances to be considered for noise contribution.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Main* |
| Field | *Instance List:* |

## generalnoiseinstonoff

```
spectre.opts generalnoiseinstonoff string any_string_value
```


© 2023 Cadence Design Systems, Inc.

### Description

Specify whether to enable ("on") or disable ("off") the noise contribution for the instance list given in the `generalnoiseinst` cdsenv.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Main* |
| Field | *Noise Contribution* |

## highvoltage

`spectre.opts highvoltage string` *`any_string_value`*

### Description

Enables optimized Spectre settings for high voltage designs including voltage, and current binning, excluding VerilogA and dangling nodes from convergence checks, and optimized large capacitance handling. Possible values are 'no' and 'yes'.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Main* |
| Field | *highvoltage* |

## iabstol

`spectre.opts iabstol string` *`any_string_value`*

### Description

Convergence criterion for absolute current tolerance.

The default is `1e-12`.


© 2023 Cadence Design Systems, Inc.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Main* |
| Field | *iabstol* |

## multithread

```
spectre.opts multithread string any_string_value
```

### Description

Enable or disable multithreading capability. When multithreading is enabled but the number of threads (nThreads) is not specified, Spectre will automatically detect the number of processors and select the proper number of threads to use.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Main* |
| Field | *multithread* |

## noiseOffType

```
spectre.opts noiseOffType toggle { thermal | flicker | shot | ign | all }
```

### Description

Disable specific noise sources for the list of instances given with the cdsenv `generalnoiseinst`.

The default is `(nil nil nil nil nil)`.

Example:

```
envSetVal("spectre.opts" "noiseOffType" 'toggle '(t nil t nil nil))
```


© 2023 Cadence Design Systems, Inc.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Main* |
| Field | *noiseoff_type* |

## noiseOnType

```
spectre.opts noiseOnType toggle { thermal | flicker | shot | ign | all }
```

### Description

Enable specific noise sources for the list of instances given with the cdsenv `generalnoiseinst`.

The default is `(nil nil nil nil nil)`.

Example:

```
envSetVal("spectre.opts" "noiseOnType" 'toggle '(t nil t nil nil))
```

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Main* |
| Field | *noiseon_type* |

## nthreads

```
spectre.opts nthreads string any_string_value
```

### Description

Specifies the number of threads for multithreading.

The default is `""`.


© 2023 Cadence Design Systems, Inc.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Main* |
| Field | *nthreads* |

## reltol

```
spectre.opts reltol string any_string_value
```

### Description

Relative convergence criterion.

The default is `1.00E-03`.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Main* |
| Field | *reltol* |

## residualtol

```
spectre.opts residualtol string any_string_value
```

### Description

Tolerance ratio for residual (multiplies reltol).

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Main* |
| Field | *residualtol* |

## temp

```
spectre.opts temp string any_string_value
```


© 2023 Cadence Design Systems, Inc.

### Description

Specifies the temperature.

The default is `"27"`.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Main* |
| Field | *temp* |

## tempeffects

`spectre.opts tempeffects string` *`any_string_value`*

### Description

Temperature effect selector. If tempeffect = vt, only thermal voltage varies with temperature; if tempeffect = tc, parameters that start with tc are active and thermal voltage is dependent on temperature; and if tempeffect = all, all built-in temperature models are enabled.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Main* |
| Field | *tempeffects* |

## tnom

`spectre.opts tnom string` *`any_string_value`*

### Description

Temperature measurement of the default component parameter

The default is `"27"`.


© 2023 Cadence Design Systems, Inc.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Main* |
| Field | *tnom* |

## vabstol

```
spectre.opts vabstol string any_string_value
```

### Description

Convergence criterion for absolute voltage tolerance.

The default is `1.00E-06`.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Main* |
| Field | *vabstol* |


© 2023 Cadence Design Systems, Inc.

# Simulator Options - Algorithm

## convdbg

`spectre.opts convdbg string` *`any_string_value`*

### Description

Specifies the option to diagnose convergence issues and identify problem areas. Possible values are "none", "status" and "detailed".

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Algorithm* |
| Field | *convdbg* |

## dc_pivot_check

`spectre.opts dc_pivot_check string` *`any_string_value`*

### Description

During DC analysis, the numeric pivoting is only performed when bad pivot is detected.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Algorithm* |
| Field | *dc_pivot_check* |

## dptran_gmethod

`spectre.opts dptran_gmethod string` *`any_string_value`*


© 2023 Cadence Design Systems, Inc.

### Description

Stamp gdev, gnode, or both in the dptran (homotopy) methods. Possible values are 'dev', 'node' and 'both'.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Algorithm* |
| Field | *dptran_gmethod* |

## gmethod

`spectre.opts gmethod string` *`any_string_value`*

### Description

Stamp gdev, gnode, or both in the homotopy methods (other than dptran).

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Algorithm* |
| Field | *gmethod* |

## gmin

`spectre.opts gmin string` *`any_string_value`*

### Description

`gmin` (conductance) is added to each nonlinear branch of the device to prevent simulation non-convergence. Large `gmin` impacts accuracy of current probe, small `gmin` may cause circuit convergence issue. For circuit that is sensitive to leakage current, it is recommended to set `gmin` to a small value or zero.

The default is `"1e-12"`.


© 2023 Cadence Design Systems, Inc.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Algorithm* |
| Field | *gmin* |

## gmin_check

```
spectre.opts gmin_check string any_string_value
```

### Description

Specifies that effect of `gmin` should be reported if significant. Possible values are 'no', 'max_v_only', 'max_only' and 'all'.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options- Algorithm* |
| Field | *gmin_check* |

## gmindc

```
spectre.opts gmindc string any_string_value
```

### Description

Minimum conductance across each non-linear device in DC analysis. If `gmindc` is not specified, the value of `gmindc` will be equal to `gmin`.

The default is `""`.


© 2023 Cadence Design Systems, Inc.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Algorithm* |
| Field | *gmindc* |

## homotopy

`spectre.opts homotopy string` *`any_string_value`*

### Description

Method used when there is no convergence on initial attempt of DC analysis. Possibly values are 'none', 'gmin', 'source', 'dptran', 'ptran', 'all'.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Algorithm* |
| Field | *homotopy* |

## icpriority

`spectre.opts icpriority string` *`any_string_value`*

### Description

Set the ic priority order. If set to netlist, the order from lowest to highest is readNS, netlist NS, readIC, netlist IC. If set to file, the order from lowest to highest is netlist NS, readNS, netlist IC readIC.

The default is `""`.


© 2023 Cadence Design Systems, Inc.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Algorithm* |
| Field | *icpriority* |

## ilimit

```
spectre.opts limit string any_string_value
```

### Description

Limiting algorithms to aid DC convergence. Possibly values are 'delta', 'log' and 'dev'.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options- Algorithm* |
| Field | *limit* |

## nonconv_topnum

```
spectre.opts nonconv_topnum string any_string_value
```

### Description

Top number of non-convergence nodes to be printed.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Algorithm* |
| Field | *nonconv_topnum* |

## pivabs

```
spectre.opts pivabs string any_string_value
```


© 2023 Cadence Design Systems, Inc.

### Description

Specifies the absolute pivot threshold.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Algorithm* |
| Field | *pivabs* |

## pivotdc

```
spectre.opts pivotdc string any_string_value
```

### Description

Use numeric pivoting on every iteration of DC analysis. Possible values are "no" and "yes".

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Algorithm* |
| Field | *pivotdc* |

## pivrel

```
spectre.opts pivrel string any_string_value
```

### Description

Specifies the relative pivot threshold.

The default is `1.00E-03`.


© 2023 Cadence Design Systems, Inc.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Algorithm* |
| Field | *pivrel* |

## preorder

```
spectre.opts preorder string any_string_value
```

### Description

Try this option when simulation runs out of memory or if the simulation is unreasonably slow for the size of your design. It controls the amount of matrix pre-ordering that is done and may lead to much fewer matrix fill-ins in some cases. Known cases include designs with very large number of small resistors and large number of behavioral instances containing voltage based equations. Possible values are "partial" and "full".

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Algorithm* |
| Field | *preorder* |

## rabsclamp

```
spectre.opts rabsclamp string any_string_value
```

### Description

When rabsclamp=value is specified, all instance resistors with absolute R<value are clamped to value.

The default is `""`.


© 2023 Cadence Design Systems, Inc.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Algorithm* |
| Field | *rabsclamp* |

## rabsshort

```
spectre.opts rabsshort string any_string_value
```

### Description

When this option is set, all fixed value resistors with absolute value of R<=rabsshort are shorted.

Default value is 0 for Spectre, and 1m for APS. Rabsshort can additionally be applied to variable resistors using the option 'short_cut_var_elem=yes'.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Algorithm* |
| Field | *rabsshort* |

## rebuild_matrix

```
spectre.opts rebuild_matrix string any_string_value
```

### Description

If set to yes, rebuild circuit matrix at the beginning of ac, dc, dcmatch, montecarlo, pz, stb, sweep, tdr, and tran analyses. This is to ensure consistent matrix ordering at the beginning of the analyses for consistent results. Notice that rebuild circuit matrix can result in performance overhead. Possible values are "no" and "yes".

The default is `""`.


© 2023 Cadence Design Systems, Inc.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Algorithm* |
| Field | *rebuild_matrix* |

## rforce

```
spectre.opts rforce string any_string_value
```

### Description

Resistance used when forcing nodesets and node-based initial conditions.

The default is `"1"`.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Algorithm* |
| Field | *rforce* |

## rthresh

```
spectre.opts rthresh string any_string_value
```

### Description

All instance resistors that have resistance smaller than global rthresh will use resistance form, unless their instance parameter or model parameter overwrites it. Note that resistance form of any resistor is set at the beginning of simulation and cannot be changed later, so altering the value of rthresh is of no use. You will have to start a new run if you want a different rthresh for your circuit

The default is `""`.


© 2023 Cadence Design Systems, Inc.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Algorithm* |
| Field | *rthresh* |

## try_fast_op

```
spectre.opts try_fast_op string any_string_value
```

### Description

Speed up the DC solution. For hard-to-converge designs, this feature fails and other methods are applied. In corner cases, this feature may have negative effects. If the DC analysis is unusually slow, the memory usage of the processes keeps increasing, or if DC analysis gets stuck even before homotopy methods start, set this option to "no".

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Algorithm* |
| Field | *try_fast_op* |


© 2023 Cadence Design Systems, Inc.

# Simulator Options - Component

## approx

```
spectre.opts approx string any_string_value
```

### Description

Specifies the use of approximate models. Difference between approximate and exact models. Possible values are 'yes' and 'no'.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Component* |
| Field | *approx* |

## auto_minductor

```
spectre.opts auto_minductor string any_string_value
```

### Description

Automatic insertion of missing mutual inductor coupling. Possible values are 'yes' and 'no'.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Component* |
| Field | *auto_minductor* |

## vth_vdsmin

```
spectre.opts ivth_vdsmin string any_string_value
```


© 2023 Cadence Design Systems, Inc.

### Description

Minimum Vds in constant current Vth calculation.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Component* |
| Field | *ivth_vdsmin* |

## ivthl

```
spectre.opts ivthl string any_string_value
```

### Description

Length offset for constant current Vth.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Component* |
| Field | *ivthl* |

## ivthn

```
spectre.opts ivthn string any_string_value
```

### Description

NMOS Vth current parameter.

The default is `""`.


© 2023 Cadence Design Systems, Inc.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Component* |
| Field | *ivthn* |

## ivthp

```
spectre.opts ivthp string any_string_value
```

### Description

PMOS Vth current parameter.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Component* |
| Field | *ivthp* |

## ivthw

```
spectre.opts ivthw string any_string_value
```

### Description

Width offset for constant current Vth.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Component* |
| Field | *ivthw* |

## imacromodels

```
spectre.opts macromodels string any_string_value
```


© 2023 Cadence Design Systems, Inc.

### Description

Determine whether circuit contains macromodels; at times, setting this parameter to yes helps improve performance

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Component* |
| Field | *macromodels* |

## maxrsd

`spectre.opts maxrsd string` *`any_string_value`*

### Description

Use approximation for drain/source parasitic resistors which are less then maxrsd. Applies to bsim3v3, bsim4 mosfet models

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Component* |
| Field | *maxrsd* |

## nport_default_passivity

`spectre.opts nport_default_passivity string` *`any_string_value`*

### Description

Check and enforce passivity of S-parameter for all nport instances. Default is `"disable"`, which means this global option has no effect. If set to a value other than `"disable"`, all nport elements in the netlist without a value for 'passivity' explicitly set, will have their 'passivity' argument set to the same value as specified in this global option. If an nport instance already has the 'passivity' option specified, the instance option will take priority if both are present.


© 2023 Cadence Design Systems, Inc.

Possible values are "no", "check", "enforce", "fit_weak_enforce", "fit_enforce" and "disable".

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Component* |
| Field | *nport_default_passivity* |

## nportcompress

```
spectre.opts nportcompress string any_string_value
```

### Description

Nport compression improves the efficiency of S-parameter simulation of large nport files when a certain percentage of the ports is unused, i.e., open or short circuited. Nport compression does not impact simulation accuracy. This option turns off compression if set to no and attempts to force compression if set to yes. If left unspecified, compression is on if N>=10 and the ratio of used ports is less than or equal to 0.8. Possible values are "no" and "yes".

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Component* |
| Field | *nportcompress* |

## nportcompressfiledir

```
spectre.opts nportcompressfiledir string any_string_value
```

### Description

The directory where the compressed nport S-parameter file is written to. If unspecified, it is stored in outdir.

The default is `""`.


© 2023 Cadence Design Systems, Inc.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Component* |
| Field | *nportcompressfiledir* |

## nportirfiledir

```
spectre.opts nportirfiledir string any_string_value
```

### Description

The directory to which the nport impulse response file will be written. If it is not specified, the file will be written to /home/<username>/.cadence/mmsim/. If a relative path is specified, the path is relative to the current working directory

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Component* |
| Field | *nportirfiledir* |

## nportirreuse

```
spectre.opts nportirreuse string any_string_value
```

### Description

Reuses impulse responses data for all `nport` instances. If set to "no", disables this feature. Possible values are "no" and "yes".

The default is `""`.


© 2023 Cadence Design Systems, Inc.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Component* |
| Field | *nportirreuse* |

## nportunusedportgmin

```
spectre.opts nportunusedportgmin string any_string_value
```

### Description

Default is 0, which leaves the port open-circuited. A small value loads open-circuited ports with a finite but large resistance. This introduces a small error in the response, but it induces losses which help obtain a passive response.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Component* |
| Field | *nportunusedportgmin* |

## nportunusedportrmin

```
spectre.opts nportunusedportrmin string any_string_value
```

### Description

Default is 0, which leaves the port short-circuited. A small value will insert a small resistance in place of short circuited ports. This introduces a small error in the response, but it induces losses which help obtain a passive response.

The default is `""`.


© 2023 Cadence Design Systems, Inc.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Component* |
| Field | *nportunusedportrmin* |

## nportbbsversion

`spectre.opts nportbbsversion string` *`any_string_value`*

### Description

Controls the BBspice version used for S-Parameter fitting. Possible values are "`1`" and "`2`". When set to 1, the BBspice version 1 is selected. If set to 2, the BBspice version 2 is selected.

The default value is:

- `2`—If the Spectre 20.1 version is used.
- `1`—If the Spectre 19.1 version is used.

### GUI Equivalent

| | |
|---|---|
| Command | *Options – Analog – Simulator Options – Component* |
| Field | *nportbbsversion* |

## nportbbsfittedfiledir

`spectre.opts nportbbsfittedfiledir string` *`any_string_value`*

### Description

Specifies the path to the directory where Spectre places the BBspice fitted S-Parameter file.

By default, this file is saved in a directory named `BBSpiceOutput`.

**Note:** You can either specify an absolute path or a relative path through this variable. In case a relative path is specified, for example, `../outfolder`, it will be considered in relation to the working directory of Spectre.


© 2023 Cadence Design Systems, Inc.

### GUI Equivalent

| | |
|---|---|
| Command | *Options – Analog – Simulator Options – Component* |
| Field | *nportbbsfittedfiledir* |

## minr

```
spectre.opts minr string any_string_value
```

### Description

All parasitic resistors inside devices less than global minr will be removed. The order of checking devices is the follows: 1. Check if resistors are smaller than local minr. If yes, check if it is a MOSFET or BJT. If it is a MOSFET, drop the resistor, if it is BJT, clamp to the minr value, and give a warning message for both cases. 2. Check global minr, All Parasitic resistors less than global minr are removed and a warning message is issued. 3. If the resistor is not removed and is smaller than 0.001, issue a warning

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Component* |
| Field | *minr* |

## scale

```
spectre.opts scale string any_string_value
```

### Description

Device instance scaling factor

The default is `"1"`.


© 2023 Cadence Design Systems, Inc.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Component* |
| Field | *scale* |

## scalefactor

```
spectre.opts scalefactor string any_string_value
```

### Description

ScaleFactor for Device Model Technology Scaling. The options parameter scalefactor enables device model providers to scale device technology independent of the design dimension scaling done by circuit designers. The resulting device instance scaling is defined by 'scale * scalefactor'. If the foundry uses a technology scale factor of 0.9 (scalefactor=0.9), and the circuit designer uses a design scale factor of 1e-6 (scale=1e-6), then the compounded scaling of the device instance dimension is 0.9e-6. Unlike options parameter scale, scalefactor cannot be used as a netlist parameter and cannot be altered or used in sweep statements.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Component* |
| Field | *scalefactor* |

## scalem

```
spectre.opts scalem string any_string_value
```

### Description

Model scaling factor.

The default is `"1"`.


© 2023 Cadence Design Systems, Inc.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Component* |
| Field | *scalem* |

## tmevthmod

```
spectre.opts tmevthmod string any_string_value
```

### Description

TSMC constant vth calculation. By default it is not activated. Possible values are "0" and "1".

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Component* |
| Field | *tmevthmod* |

## vthmod

```
spectre.opts vthmod string any_string_value
```

### Description

Vth output selector. 'std' outputs model equation Vth; 'vthcc' outputs constant current Vth and may impact simulation performance. Possible values are "std" and "vthcc".

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Component* |
| Field | *vthmod* |


© 2023 Cadence Design Systems, Inc.

# Simulator Options - Check

## checklimitdest

`spectre.opts checklimitdest string` *`any_string_value`*

### Description

Destination(s) where violations are written. Possible values are 'file', 'psf' and 'both'.

The default is `"psf"`.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Check* |
| Field | *checklimitdest* |

## checklimitfile

`spectre.opts checklimitfile string` *`any_string_value`*

### Description

File to which assert violations are written.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Check* |
| Field | *checklimitfile* |

## checklimitskipfile

`spectre.opts checklimitskipfile string` *`any_string_value`*


© 2023 Cadence Design Systems, Inc.

### Description

Specifies the file which contains the subcircuit masters or subcircuit master patterns to be skipped in device checking. Patterns can have any wildcard symbols.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Check* |
| Field | *checklimitskipfile* |

## checklimitskipsubs

```
spectre.opts checklimitskipsubs string any_string_value
```

### Description

Specifies the array of subcircuit masters or subcircuit master patterns to be skipped in device checking. Patterns can have any wildcard symbols.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Check* |
| Field | *checklimitskipsubs* |

## diagnose

```
spectre.opts diagnose string any_string_value
```

### Description

Print additional information that might help diagnose accuracy and convergence problems. Possible values are 'no', 'yes' and 'detailed'.

The default is `""`.


© 2023 Cadence Design Systems, Inc.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Check* |
| Field | *diagnose* |

## iccapcheck

```
spectre.opts iccapcheck string any_string_value
```

### Description

Check if nodes with initial conditions have capacitive path to ground. IC for the node without capacitance is treated as nodeset. Possibly values are 'no' and 'yes'.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Check* |
| Field | *iccapcheck* |

## gnshorts

```
spectre.opts ignshorts string any_string_value
```

### Description

Silently ignore shorted components. Possibly values are 'no' and 'yes'.

The default is `""`.


© 2023 Cadence Design Systems, Inc.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Check* |
| Field | *ignshorts* |

## redefinedparams

```
spectre.opts redefinedparams string any_string_value
```

### Description

Specify whether parameters can be redefined in the netlist. When set to warning or ignore, the simulator allows you to redefine parameters in the netlist. However, it honors only the last definition of the redefined parameter. Depending on the value set, the simulator displays warning messages for the redefined parameters or does not display any message. When set to error, the simulator does not allow you to redefine parameters in the netlist and displays an error message.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Check* |
| Field | *redefinedparams* |

## opptcheck

```
spectre.opts opptcheck string any_string_value
```

### Description

Checks operating point parameters against soft limits. Possible values are "no" and "yes".

The default is `""`.


© 2023 Cadence Design Systems, Inc.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Check* |
| Field | *opptcheck* |

## topcheck

```
spectre.opts topcheck string any_string_value
```

### Description

Check circuit topology for errors. Possible values are "no", "min", "full", "fixall", "errmin" and "errfull".

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Check* |
| Field | *topcheck* |


© 2023 Cadence Design Systems, Inc.

# Simulator Options - Annotation

## audit

`spectre.opts audit string` *`any_string_value`*

### Description

Print time required by various parts of the simulator. Possible values are 'no', 'brief' or 'detailed'.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Annotation* |
| Field | *audit* |

## cols

`spectre.opts cols string` *`any_string_value`*

### Description

Width of screen in characters.

The default is `"80"`.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Annotation* |
| Field | *cols* |

## colslog

`spectre.opts colslog string` *`any_string_value`*


© 2023 Cadence Design Systems, Inc.

### Description

Width of log-file in characters.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Annotation* |
| Field | *colslog* |

## debug

`spectre.opts debug string` *`any_string_value`*

### Description

Give debugging messages. Possible values are 'no' and 'yes'.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Annotation* |
| Field | *debug* |

## digits

`spectre.opts digits string` *`any_string_value`*

### Description

Number of digits used when printing numbers.

The default is `"5"`.


© 2023 Cadence Design Systems, Inc.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Annotation* |
| Field | *digits* |

## error

```
spectre.opts error string any_string_value
```

### Description

Generate error messages. Possible values are 'no' and 'yes'.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Annotation* |
| Field | *error* |

## info

```
spectre.opts info string any_string_value
```

### Description

Give informational messages. Possibly values are 'no' and 'yes'.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Annotation* |
| Field | *info* |

## inventory

```
spectre.opts inventory string any_string_value
```


© 2023 Cadence Design Systems, Inc.

### Description

Print summary of components used. Possibly values are 'brief', 'detailed' and 'no'.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Annotation* |
| Field | *inventory* |

## maxnotes

`spectre.opts maxnotes string` *`any_string_value`*

### Description

Maximum number of times a notice is issued per analysis. Note that this option has no effect on notices issued as part of parsing the netlist. Use the -maxnotes command-line option to control the number of all notices issued

The default is `"5"`.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Annotation* |
| Field | *maxnotes* |

## maxnotestologfile

`spectre.opts maxnotestologfile string` *`any_string_value`*

### Description

Maximum number of times a notice  message is printed to the log file per analysis. Note that this option has no effect on notices printed as part of parsing the netlist. Use the -maxnotestolog command-line option to control the number of all notices printed to the log file

The default is `""`.


© 2023 Cadence Design Systems, Inc.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Annotation* |
| Field | *maxnotestologfile* |

## maxwarns

```
spectre.opts maxwarns string any_string_value
```

### Description

Maximum number of times a warning message is issued per analysis. Note that this option has no effect on warnings issued as part of parsing the netlist. Use the -maxwarns command-line option to control the number of all warnings issued

The default is `"5"`.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Annotation* |
| Field | *maxwarns* |

## maxwarnstologfile

```
spectre.opts maxwarnstologfile string any_string_value
```

### Description

Maximum number of times a warning message is printed to the log file per analysis. Note that this option has no effect on warnings printed as part of parsing the netlist. Use the -maxwarnstolog command-line option to control the number of all warnings printed to the log file

The default is `""`.


© 2023 Cadence Design Systems, Inc.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Annotation* |
| Field | *maxwarnstologfile* |

## notation

`spectre.opts notation string` *`any_string_value`*

### Description

The notation to be used to display real numbers to the screen. Possibly values are 'eng', 'sci', 'float'.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Annotation* |
| Field | *notation* |

## note

`spectre.opts note string` *`any_string_value`*

### Description

Give notice messages

The default is `""`.


© 2023 Cadence Design Systems, Inc.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Annotation* |
| Field | *note* |

## narrate

```
spectre.opts narrate string any_string_value
```

### Description

Narrate the simulation

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Annotation* |
| Field | *narrate* |

## printstep

```
spectre.opts printstep string any_string_value
```

### Description

Enables Spectre to print results by equal step defined in .tran statement.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Annotation* |
| Field | *printstep* |

## simstat

```
spectre.opts simstat string any_string_value
```


© 2023 Cadence Design Systems, Inc.

### Description

Print simulation phase statistics report. Possible values are "basic" or "detailed".

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Annotation* |
| Field | *print statistics report* |

## title

`spectre.opts title string` *`any_string_value`*

### Description

Specifies the circuit title.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Annotation* |
| Field | *title* |

## warn

`spectre.opts warn string` *`any_string_value`*

### Description

Display warning messages. Possible values are "no" and "yes".

The default is `""`.


© 2023 Cadence Design Systems, Inc.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Annotation* |
| Field | *warn* |


© 2023 Cadence Design Systems, Inc.

# Simulator Options - Miscellaneous

## additionalArgs

`spectre.opts additionalArgs string` *`any_string_value`*

### Description

Specifies a string that adds options that are not supported through the ADE GUI.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Miscellaneous* |
| Field | *Additional arguments* |

## ahdllint

`spectre.opts ahdllint string` *`any_string_value`*

### Description

Specifies the Spectre command line argument to enable Verilog-A linter check. Possible values are 'warn', 'error', and 'force'. This option is not available in Spectre base. It is available only in APS or XPS.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Miscellaneous* |
| Field | *Warning type* |

## ahdllint_maxwarn

`spectre.opts ahdllint_maxwarn string` *`any_string_value`*


© 2023 Cadence Design Systems, Inc.

### Description

Specifies the maximum number of Verilog-A linter warning messages to be reported by the simulator for each message ID.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Miscellaneous* |
| Field | *Max warning* |

## ahdllint_on

`spectre.opts ahdllint_on string` *`any_string_value`*

### Description

When set to "on", enables the AHDL Linter feature.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Miscellaneous* |
| Field | *Linter check* |

## flow

`spectre.opts flow string` *`any_string_value`*

### Description

Specifies the default flow quantity.

The default is `""`.


© 2023 Cadence Design Systems, Inc.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Miscellaneous* |
| Field | *flow* |

## quantities

```
spectre.opts quantities string any_string_value
```

### Description

Print quantities. If quantities=min, the simulator prints out all defined quantities; if quantities=full, the simulator also prints a list of nodes and their quantities.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Miscellaneous* |
| Field | *quantities* |

## sensbinparam

```
spectre.opts sensbinparam string any_string_value
```

### Description

Sensitivity for binning models. Possible values are "no", "uncorrelated" and "correlated"

The default is `""`.


© 2023 Cadence Design Systems, Inc.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Miscellaneous* |
| Field | *sensbinparam* |

## sensfile

```
spectre.opts sensfile string any_string_value
```

### Description

Output sensitivity data file name.

The default is `"../psf/sens.output"`.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Miscellaneous* |
| Field | *sensfile* |

## sensfileonly

```
spectre.opts sensfileonly string any_string_value
```

### Description

Enable or disable raw output of sensitivity results. Possible values are "no" and "yes".

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Miscellaneous* |
| Field | *sensfileonly* |

## sensformat

```
spectre.opts sensformat string any_string_value
```


© 2023 Cadence Design Systems, Inc.

### Description

Format of sensitivity data. Possible values are "tabular" or "list".

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Miscellaneous* |
| Field | *sensformat* |

## senstype

```
spectre.opts senstype string any_string_value
```

### Description

Type of sensitivity being calculated. Possible values are "partial" or "normalized".

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Miscellaneous* |
| Field | *senstype* |

## value1

```
spectre.opts value1 string any_string_value
```

### Description

Specifies the default value quantity.

The default is `""`.


© 2023 Cadence Design Systems, Inc.

### GUI Equivalent

| | |
|---|---|
| Command | *Options - Analog - Simulator Options - Miscellaneous* |
| Field | *value* |


© 2023 Cadence Design Systems, Inc.

# D

# Environment Variables for Spectre Analyses Options Form

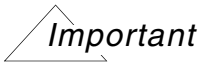
This section lists the environment variables belonging to the following partitions:

- spectre.ac
- spectre.acOpts
- spectre.dc
- spectre.dcOpts
- spectre.outputs
- spectre.relxOpts


© 2023 Cadence Design Systems, Inc.

# spectre.ac

This section describes the environment variables that you can set in the `.cdsenv` file to specify default values for Spectre AC analysis.




- The default values specified for the variables in this document indicate the default values specified for these variables in the $CDSHOME/tools/dfII/etc/tools/spectre/.cdsenv file.
- If the default value of a variable is specified as an empty string value(`""`) or as `"default"`, Spectre uses the in-built default value for that variable.

  For example, the default value for `spectre.hb noiseout` is `""`. If you do not modify this value in Virtuoso, while running the simulation, Spectre internally sets it to the in-built default value `"usb"`.

  For more details, refer *Spectre Circuit Simulator Reference Guide*.

- center
- dec
- dev
- enable
- flin_out
- incrType
- log
- mod
- out1
- outType
- rf1_freq
- rf2_freq
- rfdbm
- rfdbm3
- contriblist
- designVar
- deviceParam
- fim_out
- freq
- lin
- maxharm_nonlin
- modelParam
- out2
- rangeType
- rf1_src
- rf2_src
- rfdbm2
- rfmag


© 2023 Cadence Design Systems, Inc.

- rfmag2
- rfmag3
- span
- special
- srcType
- start
- step
- stepTypeLin
- stepTypeLog
- stop
- sweep
- useDiscrete
- useDiscrete
- useFile
- values
- valuesFile

## center

```
spectre.ac center string any_string_value
```

### Description

Specifies the center of sweep.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - ac* |
| Field | *Center* |

## contriblist

```
spectre.ac contriblist string any_string_value
```

### Description

Specifies the array of device names for distortion summary. When contriblist=[""], distortion from each non-linear device is calculated.


© 2023 Cadence Design Systems, Inc.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - ac - Specialized Analyses - Compression Distortion Summary* |
| Field | *Contributor Instances* |

## dec

`spectre.ac dec string` *`any_string_value`*

### Description

Specifies the points per decade for logarithmic sweep.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - ac - Sweep Type - Logarithmic* |
| Field | *Points Per Decade* |

## designVar

`spectre.ac designVar string` *`any_string_value`*

### Description

Specifies the name of the design variable to be swept.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - ac - Sweep Variable - Design Variable* |


© 2023 Cadence Design Systems, Inc.

| Field | *Variable Name* |
|---|---|

## dev

```
spectre.ac dev string any_string_value
```

### Description

Device instance whose parameter value is to be swept.

The default is `""`.

### GUI Equivalent

| Command | *Choosing Analyses - ac - Component Parameter* |
|---|---|
| Field | *Component Name* |

## deviceParam

```
spectre.ac deviceParam string any_string_value
```

### Description

The name of the parameter whose value is to be swept.

The default is `""`.

### GUI Equivalent

| Command | *Choosing Analyses - ac - Component Parameter* |
|---|---|
| Field | *Parameter Name* |

## enable

```
spectre.ac enable toggle { t | nil }
```

### Description


© 2023 Cadence Design Systems, Inc.

Enables the analysis on the form.

The default is `(nil)`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - ac* |
| Field | *Enabled* |

## fim_out

`spectre.ac fim_out string` *`any_string_value`*

### Description

Specifies the frequency of IM output signal for distortion summary.

The default is `"0"`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - ac - Specialized Analyses - IM2 Distortion Summary* |
| Field | *Frequency of IM Output Signal* |

## flin_out

`spectre.ac flin_out string` *`any_string_value`*

### Description

Specifies the frequency of linear output signal. This field is displayed when you select IP2, IP3, IM2 or Distortion Summary analyses. This value is used to set the start for sweep range.


© 2023 Cadence Design Systems, Inc.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - ac - Specialized Analyses* |
| Field | *Frequency of Linear Output Signal* |

## freq

`spectre.ac freq string` *`any_string_value`*

### Description

Frequency when a parameter other than frequency is being swept.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - ac* |
| Field | *At Frequency (Hz)* |

## incrType

`spectre.ac incrType string` *`any_string_value`*

### Description

Specifies the sweep type for the AC analysis. The possible values are 'Linear', 'Logarithmic' and 'Automatic'.

The default is `"Automatic"`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - ac* |
| Field | *Sweep Type* |


© 2023 Cadence Design Systems, Inc.

## lin

`spectre.ac lin string` *`any_string_value`*

### Description

Sets the number of steps for linear sweep.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - ac - Sweep Type - Linear* |
| Field | *Number of Steps* |

## log

`spectre.ac log string` *`any_string_value`*

### Description

Sets the number of steps for logarithmic sweep.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - ac - Sweep Type - Logarithmic* |
| Field | *Number of Steps* |

## maxharm_nonlin

`spectre.ac maxharm_nonlin string` *`any_string_value`*

### Description

Sets the maximum harmonics of input signal frequency induced by non-linear effect.


© 2023 Cadence Design Systems, Inc.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - ac - Specialized Analyses* |
| Field | *Maximum Non-linear Harmonics* |

## mod

```
spectre.ac mod string any_string_value
```

### Description

Specifies the model whose parameter value is to be swept.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - ac - Model Parameter* |
| Field | *Model Name* |

## modelParam

```
spectre.ac modelParam string any_string_value
```

### Description

Specifies the parameter name of the model whose value is to be swept.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - ac - Model Parameter* |
| Field | *Parameter Name* |


© 2023 Cadence Design Systems, Inc.

## out1

`spectre.ac out1 string` *`any_string_value`*

### Description

Specifies the output signal 1.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - ac - Specialized Analyses* |
| Field | *Term (Current)* and *Out+ (Voltage)* |

## out2

`spectre.ac out2 string` *`any_string_value`*

### Description

Specifies the Output signal 2 for voltage.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - ac - Specialized Analyses* |
| Field | *Out-* |

## outType

`spectre.ac outType string` *`any_string_value`*

### Description

Specifies the type of output for IP2/IP3/IM2/Distortion Summary. The possible values are 'Voltage' and 'Current'.


© 2023 Cadence Design Systems, Inc.

The default is `"Voltage"`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - ac - Specialized Analyses* |
| Field | *Output* |

## rangeType

`spectre.ac rangeType string` *`any_string_value`*

### Description

Specifies the type of sweep range for the AC analysis.

The default is `"Start-Stop"` and `"Center-Span"` .

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - ac* |
| Field | *Sweep Range* |

## rf1_freq

`spectre.ac rf1_freq string` *`any_string_value`*

### Description

Specifies the RF1 frequency when the sweep starts for IP3/IP2/IM2/DistortionSummary. This value overrides the start value specified in sweep range.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - ac - Specialized Analyses* |
| Field | *Freq* |


© 2023 Cadence Design Systems, Inc.

## rf1_src

```
spectre.ac rf1_src string any_string_value
```

### Description

Specifies the array of RF1 source names for IP3/IP2/IM2/DistortionSummary.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - ac - Specialized Analyses* |
| Field | *Input Sources 1* |

## rf2_freq

```
spectre.ac rf2_freq string any_string_value
```

### Description

Specifies the RF2 frequency when the sweep stops for IP3/IP2/IM2DistortionSummary. This value overrides the stop value specified in sweep range.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - ac - Specialized Analyses* |
| Field | *Freq* |

## rf2_src

```
spectre.ac rf2_src string any_string_value
```

### Description

Specifies the array of RF2 source names for IP3/IP2/IM2/DistortionSummary.


© 2023 Cadence Design Systems, Inc.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - ac - Specialized Analyses* |
| Field | *Input Sources 2* |

## rfdbm

```
spectre.ac rfdbm string any_string_value
```

### Description

Specifies the RF source dBm. This setting is available when you select the source type as '`port`' in the IM2, IP2 and IP3 specialized analyses.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - ac - Specialized Analyses* |
| Field | *Input Power (dBm)* |

## rfdbm2

```
spectre.ac rfdbm2 string any_string_value
```

### Description

Specifies the RF2 source dBm. This field is available for IP2 and IP3 analyses.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - ac - Specialized Analyses* |
| Field | *Power 2* |


© 2023 Cadence Design Systems, Inc.

## rfdbm3

```
spectre.ac rfdbm3 string any_string_value
```

### Description

Specifies the RF3 source dBm. This field is available for IP2 and IP3 analyses.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - ac - Specialized Analyses* |
| Field | *3* |

## rfmag

```
spectre.ac rfmag string any_string_value
```

### Description

Specifies the RF source magnitude for IP2, IP3 and IM2 analyses when the source type is 'isource' or 'vsource'.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - ac - Specialized Analyses* |
| Field | *Input Magnitude* |

## rfmag2

```
spectre.ac rfmag2 string any_string_value
```

### Description

Specifies the RF2 source magnitude. This field is available for IP2 and IP3 analyses.


© 2023 Cadence Design Systems, Inc.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - ac - Specialized Analyses* |
| Field | *Magnitude2* |

## rfmag3

```
spectre.ac rfmag3 string any_string_value
```

### Description

Specifies the RF3 source magnitude. This field is available for IP2 and IP3 analyses.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - ac - Specialized Analyses* |
| Field | *3* |

## span

```
spectre.ac span string any_string_value
```

### Description

Specifies the sweep limit span.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - ac* |
| Field | *Span* |


© 2023 Cadence Design Systems, Inc.

## special

`spectre.ac special string` *`any_string_value`*

### Description

Specifies the type of specialized AC analysis. The possible values are 'None', 'IM2 Distortion Summary', 'Compression Distortion Summary', 'Rapid IP3', and 'Rapid IP2'.

The default is `"None"`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - ac* |
| Field | *Specialized Analyses* |

## srcType

`spectre.ac srcType string` *`any_string_value`*

### Description

Specifies the source type for IP2, IP3 and IM2 specialized analyses. The possible values are port, isource and vsource.

The default is `"port"`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - ac - Specialized Analyses* |
| Field | *Source Type* |

## start

`spectre.ac start string` *`any_string_value`*

### Description


© 2023 Cadence Design Systems, Inc.

Specifies the start sweep limit.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - ac* |
| Field | *Start* |

## step

`spectre.ac step string` *`any_string_value`*

### Description

Specifies the step size for linear sweep.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - ac - Sweep Type - Linear* |
| Field | *Step Size* |

## stepTypeLin

`spectre.ac stepTypeLin string` *`any_string_value`*

### Description

Specifies the step settings for linear sweeps. The possible values are "Step Size", "Number of Steps".

The default is `"Step Size"`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - ac - Sweep Type - Linear* |


© 2023 Cadence Design Systems, Inc.

| | |
|---|---|
| Field | *Linear* |

## stepTypeLog

`spectre.ac stepTypeLog string` *`any_string_value`*

### Description

Specifies the step settings for logarithmic sweeps. The possible values are 'Points Per Decade' and 'Number of Steps'.

The default is `"Points Per Decade"`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - ac - Sweep Type - Logarithmic* |
| Field | *Logarithmic* |

## stop

`spectre.ac stop string` *`any_string_value`*

### Description

Specifies the stop sweep limit.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - ac* |
| Field | *Stop* |


© 2023 Cadence Design Systems, Inc.

## sweep

```
spectre.ac sweep string any_string_value
```

### Description

Specifies the setting for *Sweep Variable*. The possible values are 'Frequency', 'Design Variable', 'Temperature', 'Component Parameter', 'Model Parameter' and 'None'.

The default is `"Frequency"`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - ac* |
| Field | *Sweep Variable* |

## useDiscrete

```
spectre.ac useDiscrete boolean { t | nil }
```

### Description

Enables the 'Add Specific Points' check box for adding a list of points to the sweep range.

The default is `nil`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - ac* |
| Field | *Add Specific Points* |

## useFile

```
spectre.ac useFile boolean { t | nil }
```

### Description

Enables the check box adjacent to 'Add Points by File'.


© 2023 Cadence Design Systems, Inc.

The default is `nil`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - ac* |
| Field | *Add Points by File* |

## values

`spectre.ac values string` *`any_string_value`*

### Description

Specifies the array of sweep values. This field is displayed when you enable the setting *Add Specific Points*.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - ac* |
| Field | *Add Specific Points* |

## valuesFile

`spectre.ac valuesFile string` *`any_string_value`*

### Description

Specifies the name of the file containing the sweep values. This field is displayed when you enable the setting *Add Points by File*.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - ac* |

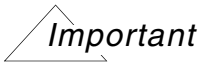

© 2023 Cadence Design Systems, Inc.

Field *Add Points by File*


© 2023 Cadence Design Systems, Inc.

# spectre.acOpts

This section describes the environment variables that you can set in the `.cdsenv` file to specify default values for Spectre AC analysis options.




- The default values specified for the variables in this document indicate the default values specified for these variables in the $CDSHOME/tools/dfII/etc/tools/spectre/.cdsenv file.
- If the default value of a variable is specified as an empty string value(`""`) or as `"default"`, Spectre uses the in-built default value for that variable.

  For example, the default value for `spectre.hb noiseout` is `""`. If you do not modify this value in Virtuoso, while running the simulation, Spectre internally sets it to the in-built default value `"usb"`.

  For more details, refer *Spectre Circuit Simulator Reference Guide*.

- additionalParams
- annotate
- force
- nestlvl
- oppoint
- prevoppoint
- readforce
- readns
- restart
- save
- skipdc
- write
- writefinal
-


© 2023 Cadence Design Systems, Inc.

## additionalParams

```
spectre.acOpts additionalParams string any_string_value
```

### Description

Specifies any additional parameters required for the AC analysis.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - ac - Options* |
| Field | *additionalParams* |

## annotate

```
spectre.acOpts annotate string any_string_value
```

### Description

Specifies the degree of annotation. Possible values are 'no', 'title', 'sweep', 'status' and 'steps'.

The default is `"status"`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - ac - Options* |
| Field | *annotate* |

## force

```
spectre.acOpts force string any_string_value
```

### Description

Specifies the set of initial conditions to use. Possible values are 'none', 'node', 'dev' and 'all'.


© 2023 Cadence Design Systems, Inc.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - ac - Options* |
| Field | *force* |

## nestlvl

```
spectre.acOpts nestlvl string any_string_value
```

### Description

Specifies the levels of subcircuits to output.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - ac - Options* |
| Field | *nestlvl* |

## oppoint

```
spectre.acOpts oppoint string any_string_value
```

### Description

Determines whether operating point information should be computed:if yes, where should it be printed (screen or file). Operating point information is not printed if the operating point computed in the previous analysis remains unchanged. Possible values are 'no', 'screen', 'logfile', and 'rawfile'.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - ac - Options* |


© 2023 Cadence Design Systems, Inc.

| Field | *oppoint* |
|---|---|

## prevoppoint

`spectre.acOpts prevoppoint string` *`any_string_value`*

### Description

Specifies if the operating point computed in the previous analysis is used. Possible values are 'no' and 'yes'.

The default is `""`.

### GUI Equivalent

| Command | *Choosing Analyses - ac - Options* |
|---|---|
| Field | *prevoppoint* |

## readforce

`spectre.acOpts readforce string` *`any_string_value`*

### Description

Specifies the file that contains initial conditions.

The default is `""`.

### GUI Equivalent

| Command | *Choosing Analyses - ac - Options* |
|---|---|
| Field | *readforce* |


© 2023 Cadence Design Systems, Inc.

## readns

`spectre.acOpts readns string` *`any_string_value`*

### Description

Specifies the file that contains an estimate of the DC solution (nodeset).

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - ac - Options* |
| Field | *readns* |

## restart

`spectre.acOpts restart string` *`any_string_value`*

### Description

Restarts the DC solution from scratch if any condition has changed. If not, uses the previous solution as an initial guess. Possible values are 'no' and 'yes'.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - ac - Options* |
| Field | *restart* |

## save

`spectre.acOpts save string` *`any_string_value`*

### Description

Specifies the signals to output. Possible values are 'all', 'lvl', 'allpub', 'lvlpub', and 'selected'.


© 2023 Cadence Design Systems, Inc.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - ac - Options* |
| Field | *save* |

## skipdc

`spectre.acOpts skipdc string` *`any_string_value`*

### Description

Skips the DC analysis. Possible values are 'no' and 'yes'.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - ac - Options* |
| Field | *skipdc* |

## write

`spectre.acOpts write string` *`any_string_value`*

### Description

Specifies the DC operating point output file at the first step of the sweep.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - ac - Options* |
| Field | *write* |


© 2023 Cadence Design Systems, Inc.

## writefinal

`spectre.acOpts writefinal string` *`any_string_value`*

### Description

Specifies the DC operating point output file at the last step of the sweep.

The default is `""`.

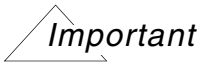
### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - ac - Options* |
| Field | *writefinal* |


© 2023 Cadence Design Systems, Inc.

# spectre.dc

This section describes the environment variables that you can set in the `.cdsenv` file to specify default values for Spectre AC analysis options.




- The default values specified for the variables in this document indicate the default values specified for these variables in the $CDSHOME/tools/dfII/etc/tools/spectre/.cdsenv file.
- If the default value of a variable is specified as an empty string value(`""`) or as `"default"`, Spectre uses the in-built default value for that variable.

  For example, the default value for `spectre.hb noiseout` is `""`. If you do not modify this value in Virtuoso, while running the simulation, Spectre internally sets it to the in-built default value `"usb"`.

  For more details, refer *Spectre Circuit Simulator Reference Guide*.


- center
- designVar
- deviceParam
- hysteresis
- lin
- mod
- rangeType
- span
- step
- stepTypeLog
- sweep
- useFile
- valuesfile
- dec
- dev
- enable
- incrType
- log
- modelParam
- saveOppoint
- start
- stepTypeLin
- stop
- useDiscrete
- values



© 2023 Cadence Design Systems, Inc.

## center

`spectre.dc center string` *`any_string_value`*

### Description

Specifies the center of sweep.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - dc - Sweep Range - Center-Span* |
| Field | *Center* |

## dec

`spectre.dc dec string` *`any_string_value`*

### Description

Specifies the points per decade.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - dc - Sweep Type - Logarithmic* |
| Field | Points per Decade |

## designVar

`spectre.dc designVar string` *`any_string_value`*

### Description

Specifies the name of the design variable to be swept.


© 2023 Cadence Design Systems, Inc.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - dc - Sweep Variable - Design Variable* |
| Field | *Variable Name* |

## dev

```
spectre.dc dev string any_string_value
```

### Description

Device instance whose parameter value is to be swept.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - dc - Sweep Variable - Component Parameter* |
| Field | *Component Name* |

## deviceParam

```
spectre.dc deviceParam string any_string_value
```

### Description

The name of the parameter whose value is to be swept.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - dc - Sweep Variable - Component Parameter* |


© 2023 Cadence Design Systems, Inc.

| Field | *Parameter Name* |
|---|---|

## enable

```
spectre.dc enable toggle { t | nil }
```

### Description

Enables the analysis on the form. Possible values are (t) or (nil).

The default is `(nil)`.

### GUI Equivalent

| Command | *Choosing Analyses - dc* |
|---|---|
| Field | *Enabled* |

## hysteresis

```
spectre.dc hysteresis boolean { t | nil }
```

### Description

Perform DC hysteresis sweep. When set to yes, a reverse sweep will automatically be added to the DC sweep.

The default is `nil`.

### GUI Equivalent

| Command | *Choosing Analyses - dc - DC Analysis* |
|---|---|
| Field | *Hysteresis Sweep* |


© 2023 Cadence Design Systems, Inc.

## incrType

`spectre.dc incrType string` *`any_string_value`*

### Description

Specifies the sweep type for the DC analysis. This is available only when you select a Sweep variable. Possible values are 'Linear', 'Logarithmic' and 'Automatic'.

The default is `"Automatic"`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - dc - Sweep Variable* |
| Field | *Sweep Type* |

## lin

`spectre.dc lin string` *`any_string_value`*

### Description

Specifies the number of steps in linear sweep.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - dc - Sweep Type - Linear* |
| Field | *Number of Steps* |

## log

`spectre.dc log string` *`any_string_value`*

### Description

Specifies the number of steps in logarithmic sweep.


© 2023 Cadence Design Systems, Inc.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - dc - Sweep Type - Logarithmic* |
| Field | *Number of Steps* |

## mod

`spectre.dc mod string` *`any_string_value`*

### Description

Model whose parameter value is to be swept.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - dc - Sweep Variable - Model Parameter* |
| Field | *Model Name* |

## modelParam

`spectre.dc modelParam string` *`any_string_value`*

### Description

Specifies the parameter name of the model whose value is to be swept.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - dc - Sweep Variable - Model Parameter* |


© 2023 Cadence Design Systems, Inc.

| Field | *Parameter Name* |
|---|---|

## rangeType

```
spectre.dc rangeType string any_string_value
```

### Description

Specifies the type of sweep range for the DC analysis. This is available when you select a Sweep Variable. Possible values are 'Start-Stop' and 'Center-Span'.

The default is `"Start-Stop"`.

### GUI Equivalent

| Command | *Choosing Analyses - dc* |
|---|---|
| Field | *Sweep Range* |

## saveOppoint

```
spectre.dc saveOppoint boolean { t | nil }
```

### Description

Enables or disables the *Save DC Operating Point* check box in the DC analysis setup form and adds an `oppoint` info analysis to the netlist.

The default is `nil`.

### GUI Equivalent

| Command | *Choosing Analyses - dc* |
|---|---|
| Field | *Save DC Operating Point* |


© 2023 Cadence Design Systems, Inc.

## span

`spectre.dc span string` *`any_string_value`*

### Description

Specifies the sweep limit span.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - dc - Sweep Range - Center-Span* |
| Field | *Span* |

## start

`spectre.dc start string` *`any_string_value`*

### Description

Specifies the start sweep limit.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - dc - Sweep Range - Start-Stop* |
| Field | *Start* |

## step

`spectre.dc step string` *`any_string_value`*

### Description

Specifies the step size for linear sweep.


© 2023 Cadence Design Systems, Inc.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - dc - Sweep Type - Linear* |
| Field | *Step Size* |

## stepTypeLin

`spectre.dc stepTypeLin string` *`any_string_value`*

### Description

Specifies the step settings for linear sweeps. The possible values are "Step Size", "Number of Steps".

The default is `"Step Size"`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - dc - Sweep Type - Linear* |
| Field | *Linear* |

## stepTypeLog

`spectre.dc stepTypeLog string` *`any_string_value`*

### Description

Specifies the step settings for logarithmic sweeps. The possible values are 'Points Per Decade' and 'Number of Steps'.

The default is `"Points Per Decade"`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - dc - Sweep Type - Logarithmic* |


© 2023 Cadence Design Systems, Inc.

| Field | *Logarithmic* |
|---|---|

## stop

```
spectre.dc stop string any_string_value
```

### Description

Specifies the stop sweep limit.

The default is `""`.

### GUI Equivalent

| Command | *Choosing Analyses - dc - Sweep Range - Start-Stop* |
|---|---|
| Field | *Stop* |

## sweep

```
spectre.dc sweep string any_string_value
```

### Description

Specifies the setting for *Sweep Variable*. The possible values are 'Design Variable', 'Temperature', 'Component Parameter', and 'Model Parameter'.

The default is `""`.

### GUI Equivalent

| Command | *Choosing Analyses - dc* |
|---|---|
| Field | *Sweep Variable* |


© 2023 Cadence Design Systems, Inc.

## useDiscrete

```
spectre.dc useDiscrete boolean { t | nil }
```

### Description

Enables the 'Add Specific Points' check box for adding a list of points to the sweep range.

The default is `nil`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - dc* |
| Field | *Add Specific Points* |

## useFile

```
spectre.dc useFile boolean { t | nil }
```

### Description

Enables the 'Add Points By File' check box for specifying the file that contains the sweep values.

The default is `nil`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - dc* |
| Field | *Add Points By File* |

## values

```
spectre.dc values string any_string_value
```

### Description

Specifies an array of specific sweep values.


© 2023 Cadence Design Systems, Inc.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - dc* |
| Field | *Add Specific Points* |

## valuesfile

`spectre.dc valuesfile string` *`any_string_value`*

### Description

Specifies the file that contains the sweep values.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - dc* |
| Field | *Add Points By File* |


© 2023 Cadence Design Systems, Inc.

# spectre.dcOpts

This section describes the environment variables that you can set in the `.cdsenv` file to specify default values for Spectre AC analysis options.

*Important*

- The default values specified for the variables in this document indicate the default values specified for these variables in the $CDSHOME/tools/dfII/etc/tools/spectre/.cdsenv file.
- If the default value of a variable is specified as an empty string value(`""`) or as `"default"`, Spectre uses the in-built default value for that variable.

  For example, the default value for `spectre.hb noiseout` is `""`. If you do not modify this value in Virtuoso, while running the simulation, Spectre internally sets it to the in-built default value `"usb"`.

  For more details, refer *Spectre Circuit Simulator Reference Guide*.


- additionalParams
- annotate
- captab
- check
- detail
- emirfile
- emirformat
- filename
- force
- homotopy
- maxiters
- maxsteps
- nestlvl
- oppoint
- print
- readforce
- readns
- restart
- save
- saveopinfinal
- sort
- threshold
- write
- writefinal



© 2023 Cadence Design Systems, Inc.

## additionalParams

`spectre.dcOpts additionalParams string` *`any_string_value`*

### Description

Specifies any additional parameters required for the DC analysis.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - dc - Options - ADDITIONAL PARAMETERS* |
| Field | *additionalParams* |

## annotate

`spectre.dcOpts annotate string` *`any_string_value`*

### Description

Specifies the degree of annotation. Possible values are 'no', 'title', 'sweep', 'status', and 'steps'.

The default is `"status"`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - dc - Options - ANNOTATION PARAMETERS* |
| Field | *annotate* |


© 2023 Cadence Design Systems, Inc.

## captab

```
spectre.dcOpts captab boolean { t | nil }
```

### Description

Enables the writing of captab information data when DC Operating Point analysis is run.

The default is `nil`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - dc - Options - CAPTAB PARAMETERS* |
| Field | *captab* |

## check

```
spectre.dcOpts check string any_string_value
```

### Description

Check s the operating point parameters against soft limits. Possible values are 'yes' and 'no'. The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - dc - Options - OUTPUT PARAMETERS* |
| Field | *check* |


© 2023 Cadence Design Systems, Inc.

## detail

`spectre.dcOpts detail string` *`any_string_value`*

### Description

Specifies how detailed the capacitance table should be if a captab analysis is enabled. Possible values are 'node', 'nodetoground', and 'nodetonode'.

The default is `"node"`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - dc - Options - CAPTAB PARAMETERS* |
| Field | *detail* |

## emirfile

`spectre.dcOpts emirfile string` *`any_string_value`*

### Description

Specifies the name of the EM/IR database file to be generated if enabled.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - dc - Options - EM/IR OUTPUT PARAMETERS* |
| Field | *emirfile* |


© 2023 Cadence Design Systems, Inc.

## emirformat

`spectre.dcOpts emirformat string` *`any_string_value`*

### Description

Specifies the format of the EM/IR database file. Possible values are 'none' and 'vavo'.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - dc - Options - EM/IR OUTPUT PARAMETERS* |
| Field | *emirformat* |

## filename

`spectre.dcOpts filename string` *`any_string_value`*

### Description

Displays the file name when you set oppoint to 'file'.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - dc - Options* |
| Field | *filename* |


© 2023 Cadence Design Systems, Inc.

## force

```
spectre.dcOpts force string any_string_value
```

### Description

Determine whether to force values for DC. Uses the values from the device and node ICs. Possible values are 'none', 'node', ‘dev', and 'all’.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - dc - Options - INITIAL CONDITION PARAMETERS* |
| Field | *force* |

## homotopy

```
spectre.dcOpts homotopy string any_string_value
```

### Description

Method used when no convergence occurs on initial attempt of DC analysis. Possible values are 'gmin’, 'source', 'dptran', 'ptran', 'none', and 'all'.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - dc - Options - CONVERGENCE PARAMETERS* |
| Field | *homotopy* |


© 2023 Cadence Design Systems, Inc.

## maxiters

`spectre.dcOpts maxiters string` *`any_string_value`*

### Description

Specifies the maximum number of iterations.

The default is `"150"`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - dc - Options - CONVERGENCE PARAMETERS* |
| Field | *maxiters* |

## maxsteps

`spectre.dcOpts maxsteps string` *`any_string_value`*

### Description

Specifies the maximum number of steps used in homotopy method.

The default is `"10000"`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - dc - Options - CONVERGENCE PARAMETERS* |
| Field | *maxsteps* |


© 2023 Cadence Design Systems, Inc.

## nestlvl

```
spectre.dcOpts nestlvl string any_string_value
```

### Description

Specifies the levels of subcircuits to output. This is enabled when you set 'save'='lvlpub' or 'lvl'.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - dc - Options - OUTPUT PARAMETERS* |
| Field | *nestlvl* |

## oppoint

```
spectre.dcOpts oppoint string any_string_value
```

### Description

Specifies if operating point information is computed; if yes, where it is printed (screen or file). Possible values are 'no', 'rawfile', screen', 'logfile', and 'file'.

The default is `"rawfile"`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - dc - Options - OUTPUT PARAMETERS* |
| Field | *oppoint* |


© 2023 Cadence Design Systems, Inc.

## print

```
spectre.dcOpts print string any_string_value
```

### Description

Prints the node voltages. Possible values are 'yes' and 'no'.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - dc - Options - OUTPUT PARAMETERS* |
| Field | *print* |

## readforce

```
spectre.dcOpts readforce string any_string_value
```

### Description

Specifies the file that contains the force values.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - dc - Options - INITIAL CONDITION PARAMETERS* |
| Field | *readforce* |


© 2023 Cadence Design Systems, Inc.

## readns

`spectre.dcOpts readns string` *`any_string_value`*

### Description

Specifies the file that contains the estimates of DC solution (nodeset).

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - dc - Options - STATE-FILE PARAMETERS* |
| Field | *readns* |

## restart

`spectre.dcOpts restart string` *`any_string_value`*

### Description

Restarts from scratch if any condition has changed. If no changes are found, uses the previous solution as initial guess. Possible values are 'no' and 'yes'.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - dc - Options - CONVERGENCE PARAMETERS* |
| Field | *restart* |


© 2023 Cadence Design Systems, Inc.

## save

```
spectre.dcOpts save string any_string_value
```

### Description

Specifies the signals to output. Possible values are 'selected', 'lvlpub', 'lvl', 'allpub', and ‘all'.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - dc - Options - OUTPUT PARAMETERS* |
| Field | *save* |

## saveopinfinal

```
spectre.dcOpts saveopinfinal boolean { t | nil }
```

### Description

Specifies whether the DC Operating Point analysis is performed before the DC sweep (saveopinfinal=nil) or after (saveopinfinal=t). It is used for changing the netlist order.

The default is `nil`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - dc - Options - STATE-FILE PARAMETERS* |
| Field | *save op in final* |


© 2023 Cadence Design Systems, Inc.

## sort

`spectre.dcOpts sort string` *`any_string_value`*

### Description

Specifies the criteria for sorting captab parameters. Possible values are 'name' and 'value'.

The default is `"name"`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - dc - Options - CAPTAB PARAMETERS* |
| Field | *sort* |

## threshold

`spectre.dcOpts threshold string` *`any_string_value`*

### Description

Specifies the threshold capacitance value for the captab analysis.

The default is `"0.0"`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - dc - Options - CAPTAB PARAMETERS* |
| Field | *threshold* |


© 2023 Cadence Design Systems, Inc.

## write

```
spectre.dcOpts write string any_string_value
```

### Description

Specifies the file to which the solution is written at the first step of the sweep.

The default is `"spectre.dc"`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - dc - Options - STATE-FILE PARAMETERS* |
| Field | *write* |

## writefinal

```
spectre.dcOpts writefinal string any_string_value
```

### Description

Specifies the file to which the solution is written at the last step of the sweep.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Choosing Analyses - dc - Options - STATE-FILE PARAMETERS* |
| Field | *writefinal* |


© 2023 Cadence Design Systems, Inc.

# spectre.outputs

## subcktoppoint

```
spectre.outputs subcktoppoint boolean { t | nil }
```

### Description

Sets the *Schematic Pcell OP Parameters to be Saved* check box as the default selection.

The default is `"t"`.

### GUI Equivalent

| | |
|---|---|
| Command | *Outputs - Save All- Save Options* form- *Save By Subckt- Schematic Pcell OP Parameters to be Saved* |
| Field | *Schematic Pcell OP Parameters to be Saved* |

## savePcell

```
spectre.outputs savePcell string any_string_value
```

### Description

Populates the string field with the desired parameters by default.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Outputs - Save All- Save Options* form- *Save By Subckt- Schematic Pcell OP Parameters to be Saved* |
| Field | *Schematic Pcell OP Parameters to be Saved* |


© 2023 Cadence Design Systems, Inc.

## saveBySubcktInstances

```
spectre.outputs saveBySubcktInstances string
     "<Click_here_to_add_an_Instance>;;;;;;;;all;;" nil
```

### Description

Specifies a list of default values for the columns in the *Save by Subckt Instances* table on the *Save By Subckt* tab of the *Save Options* form. These default values for columns are shown in each new row added to the table.

The default value is `"<Click_here_to_add_an_Instance>;;;;;;;;all;;"`.

### Valid Value

A semi-colon separated list of nine values in string format..

### GUI Equivalent

| | |
|---|---|
| Command | *Outputs - Save All- Save Options* form- *Save By Subckt- Save By Subckt Instances* |
| Field | *Save By Subckt Instances* |


© 2023 Cadence Design Systems, Inc.

# spectre.relxOpts


- agingPointsType
- checknegagingclamp
- degCheckException
- degSortValue
- enableOutputDevDegInFile
- enableSelfheating
- gaAgePointsUnit
- gaAgeStepStop
- gaAgeStepTotalStep
- gaMode
- gaSaveSome
- mosAgeDomain
- mosAgeModelFilePath
- mosAgingTimeUnit
- mosDeltad
- mosHciIdMethod
- mosMaskDev
- mosMaskDevInstance
- mosMode
- otherRXOptions
- relxTranStop
- uriDebugMode
- uriMode
- checknegagingtype
- degSortType
- enableDegSort
- enableRelXTran
- gaAgePoints
- gaAgeStepStart
- gaAgeStepType
- gaAgeStepUnit
- gaSaveResults
- mindtemp
- mosAgeMethod
- mosAgingTime
- mosCalcLifetime
- mosHciAccuracy
- mosHciOpMethod
- mosMaskDevInclude
- GUI Equivalent
- mosOutputDevDegInFilePath
- relxTranStart
- reportModelParamChanged
- uriLibs


## agingPointsType

```
spectre.relxOpts agingPointsType string { Standard aging | Gradual aging }
```

### Description


© 2023 Cadence Design Systems, Inc.

Specifies the aging type for reliability simulation. Possible values are `Standard aging` and `Gradual aging`.

The default is `Standard aging`.

### GUI Equivalent

| | |
|---|---|
| Command | *Reliability Options – Basic – Aging Options* |
| Fields | ■ `Standard aging`: *Single Age Point*<br>■ `Gradual aging`: *Gradual Aging* |

## checknegagingclamp

```
spectre.relxOpts checknegagingclamp string { yes | no }
```

### Description

Specifies whether to clamp the degradation values for negative aging. Possible values are `yes` and `no`.

The default is `no`.

### GUI Equivalent

| | |
|---|---|
| Command | *Reliability Options – Degradation – Miscellaneous* |
| Field | *Clamp Degradation Values for Negative Aging* |

## checknegagingtype

```
spectre.relxOpts checknegagingtype string { error | warn | ignore}
```

### Description

Specifies the type of message to be generated when negative aging occurs. Possible values are `error`, `warn`, and `ignore`.


© 2023 Cadence Design Systems, Inc.

The default is `error`.

### GUI Equivalent

| | |
|---|---|
| Command | *Reliability Options – Degradation – Miscellaneous* |
| Field | *Negative Aging Check Level* |

## degCheckException

`spectre.relxOpts degCheckException string` *`a_string_value`*

### Description

Specifies the devices to be excluded from the degradation check when performing reliability analysis.

The default is `" "`.

### GUI Equivalent

| | |
|---|---|
| Command | *Reliability Options – Degradation – Device Degradation* |
| Field | *Devices to be Excluded from Degradation check* |

## degSortType

`spectre.relxOpts degSortType string { threshold | number }`

### Description

Specifies the method to filter the degradation results.

Possible values are:

- `threshold`: Prints devices having degradation values greater than the specified threshold value.
- `number`: Prints only the specified number of devices that have the highest degradation.


© 2023 Cadence Design Systems, Inc.

The default value is `threshold`.

### GUI Equivalent

| | |
|---|---|
| Command | *Reliability Options – Output – Filtering – Filter Criteria* |
| Field | ■ `threshold`: *Degradation Threshold*<br>■ `number`: *Max Devices* |

## degSortValue

`spectre.relxOpts degSortValue string` *`a_string_value`*

### Description

Depending upon the method set by the `degSortType` environment variable, specifies either the degradation threshold value or the maximum number of devices based on which the degradation results are filtered.

The table shows how the degradation results are filtered for the following values of the `degSortType` and `degSortValue` environment variables:

| degSortType | degSortValue | Output |
|---|---|---|
| `threshold` | `0.1` | Devices that have degraded by more than 10% are printed in the output. |
| `number` | `20` | 20 devices that have degraded the most are printed in the output in the descending order of the degradation value. |

The default is `" "`.

### GUI Equivalent

| | |
|---|---|
| Command | *Reliability Options – Output – Filtering – Filter Criteria* |
| Field | ■ The text field next to the *Degradation Threshold* option<br>■ The text field next to the *Max Devices* option |


© 2023 Cadence Design Systems, Inc.

## enableDegSort

```
spectre.relxOpts enableDegSort boolean { t | nil }
```

### Description

Specifies whether to filter the degradation results. Possible values are `t` and `nil`.

The default is `nil`.

**Note:** After enabling the filtering of the degradation results, you can specify the filtering criteria by specifying the `degSortType` environment variable.

### GUI Equivalent

| | |
|---|---|
| Command | *Reliability Options – Output – Filtering* |
| Field | *Filter Degradation Results in Report* |

## enableOutputDevDegInFile

```
spectre.relxOpts enableOutputDevDegInFile string { File | GUI }
```

### Description

Specifies the method to be used for calculating device degradation values. Possible values are `File` and `GUI`.

The default is `GUI`.

### GUI Equivalent

| | |
|---|---|
| Command | *Reliability Options – Degradation – Device Degradation* |
| Field | ■ `GUI`: *GUI*<br>■ `File`: *File* |


© 2023 Cadence Design Systems, Inc.

## enableRelXTran

```
spectre.relxOpts enableRelXTran boolean { t | nil }
```

### Description

Specifies whether to define the transient time window for the stress test. Possible values are `t` and `nil`.

The default is `nil`.

**Note:** After setting this environment variable to `t`, you can use the `relxTranStart` and `relxTranStop` environment variables to define the start and stop time of transient time window for the stress test.

### GUI Equivalent

| | |
|---|---|
| Command | *Reliability Options – Basic – Stress Test* |
| Field | *Transient Time Window for Stress Test* |

## enableSelfheating

```
spectre.relxOpts enableSelfheating boolean { t | nil }
```

### Description

Specifies whether to use the self-heating effect to obtain the reliability values.

The default is `nil`.

### GUI Equivalent

| | |
|---|---|
| Command | *Reliability Options – Basic – Analysis* |
| Field | *Self-Heating* |


© 2023 Cadence Design Systems, Inc.

## gaAgePoints

```
spectre.relxOpts gaAgePoints string a_string_value
```

### Description

Specifies the age points for the gradual aging simulation.

The default is `10`.

**Note:** You can specify more than one age points in a list of space-separated values. For example:

```
envSetVal("spectre.relxOpts" "gaAgePoints" 'string "10 20 30")
```

### GUI Equivalent

| | |
|---|---|
| Command | *Reliability Options – Basic – Aging Options – Gradual Aging – Mode (Age Points)* |
| Field | *Age* |

## gaAgePointsUnit

```
spectre.relxOpts gaAgePointsUnit string { Years | Days | Hours | Minutes | Seconds }
```

### Description

Specifies the unit of the age points in gradual aging simulation. Possible values are `Years`, `Days`, `Hours`, `Minutes` and `Seconds`.

The default is `Years`.

### GUI Equivalent

| | |
|---|---|
| Command | *Reliability Options – Basic – Aging Options – Gradual Aging – Mode (Age Points)* |
| Field | *Age Unit* |


© 2023 Cadence Design Systems, Inc.

## gaAgeStepStart

```
spectre.relxOpts gaAgeStepStart string a_string_value
```

### Description

Specifies the start time of an age step in the gradual aging simulation.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Reliability Options – Basic – Aging Options – Gradual Aging – Mode (Age Sweep)* |
| Field | *Start Time* |

## gaAgeStepStop

```
spectre.relxOpts gaAgeStepStop string a_string_value
```

### Description

Specifies the stop time of an age step in the gradual aging simulation.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Reliability Options – Basic – Aging Options – Gradual Aging – Mode (Age Sweep)* |
| Field | *Stop Time* |

## gaAgeStepTotalStep

```
spectre.relxOpts gaAgeStepTotalStep string a_string_value
```

### Description


© 2023 Cadence Design Systems, Inc.

Specify the total number of age steps in the gradual aging simulation.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Reliability Options – Basic – Aging Options – Gradual Aging – Mode (Age Sweep)* |
| Field | *Total Steps* |

## gaAgeStepType

```
spectre.relxOpts gaAgeStepType string { lin | log }
```

### Description

Specifies the age step method for the *Age Sweep* mode of the gradual aging flow. Possible values are:

- `lin`: Use the linear method to calculate the time step. If you select this method, the time step is calculated using the following formula:

  ```
  (Stop Time - Start Time) / (Total Steps -1)
  ```

- `log`: Use the logarithm model to calculate the time step. If you select this method, the time step is calculated using the following formula:

  ```
  [log(Stop Time - Start Time)] / (Total Steps -1)
  ```

The default is `lin`.

### GUI Equivalent

| | |
|---|---|
| Command | *Reliability Options – Basic – Aging Options – Gradual Aging – Mode (Age Sweep)* |
| Field | *Type* |


© 2023 Cadence Design Systems, Inc.

## gaAgeStepUnit

```
spectre.relxOpts gaAgeStepUnit string { Years | Days | Hours | Minutes | Seconds }
```

### Description

Specifies the unit to measure the start and stop time for each age step. Possible values are `Years`, `Days`, `Hours`, `Minutes` and `Seconds`.

The default is `Years`.

### GUI Equivalent

| | |
|---|---|
| Command | *Reliability Options – Basic – Aging Options – Gradual Aging – Mode (Age Sweep)* |
| Field | *Unit* |

## gaMode

```
spectre.relxOpts gaMode string { Age points | Iteration | Age sweep}
```

### Description

Specifies the mode of gradual aging simulation. Possible values are `Age points`, `Iteration`, and `Age sweep`.

The default is `Age Points`.

### GUI Equivalent

| | |
|---|---|
| Command | *Reliability Options – Basic – Aging Options – Gradual Aging – Mode* |
| Field | ■ `Age points`: *Age Points*<br>■ `Age sweep`: *Age Sweep* |


© 2023 Cadence Design Systems, Inc.

## gaSaveResults

```
spectre.relxOpts gaSaveResults string { all | none | some }
```

### Description

Specifies the age points for which gradual aging results are to be saved. Possible values are:

- `all`: Saves the gradual aging results for all the age points.
- `none`: Saves the gradual aging results only for the last age point.
- `some`: Saves the gradual aging results for the age points based on the index values specified by the `gaSaveSome` environment variable.

The default is `all`.

### GUI Equivalent

| | |
|---|---|
| Command | *Reliability Options – Basic – Aging Options – Gradual Aging* |
| Field | *Save Intermediate Results* |

## gaSaveSome

```
spectre.relxOpts gaSaveSome string a_string_value
```

### Description

Specifies the index values of the age points for which you want to save the gradual aging results. You can specify any integer value greater than zero as the index.

The default is `""`.

The index value for the first age point starts with `1`. If you have set the `gaAgePoints` environment variable to `envSetVal("spectre.relxOpts" "gaAgePoints" 'string "10 15 20 25 30")`, and want to save the gradual aging results for the intermediate age points `15` and `25`, set the `gaSaveSome` environment variable to the following:

```
envSetVal("spectre.relxOpts" "gaSaveSome" 'string "2 4")
```


© 2023 Cadence Design Systems, Inc.

**Note:** The gradual aging results for the last age point are always saved by default.

### GUI Equivalent

| | |
|---|---|
| Command | *Reliability Options – Basic – Aging Options – Gradual Aging – Save Intermediate Results (Some)* |
| Field | The text field next to the *Some* option |

## mindtemp

```
spectre.relxOpts mindtemp string a_string_value
```

### Description

Specifies the minimum delta temperature for self-heating analysis.

The default is `""`.

### GUI Equivalent

None

## mosAgeDomain

```
spectre.relxOpts mosAgeDomain string { loglog | linlin | linlog}
```

### Description

Specifies the domain of calculation to be used in the interpolation or regression age method.

The default is `loglog`.

### GUI Equivalent

| | |
|---|---|
| Command | *Reliability Options (RelXpert) – Modeling – Age Modeling – Age Method* |
| Field | *Scale* |


© 2023 Cadence Design Systems, Inc.

## mosAgeMethod

```
spectre.relxOpts mosAgeMethod string { interp | regres | agemos}
```

### Description

Specifies the method for calculating degraded SPICE model parameters for aged circuit simulation. Possible values are `interp`, `regres`, and `agemos`.

The default is `agemos`.

### GUI Equivalent

| | |
|---|---|
| Command | *Reliability Options (RelXpert) – Modeling – Age Modeling – Age Method* |
| Field | *Method* |

## mosAgeModelFilePath

```
spectre.relxOpts mosAgeModelFilePath string a_string_value
```

### Description

Specifies the path of the file in which the degraded models are to be dumped.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Reliability Options – Output – Reports* |
| Field | *Save Models for Aged Devices* |

## mosAgingTime

```
spectre.relxOpts mosAgingTime string any_string_value
```

### Description


© 2023 Cadence Design Systems, Inc.

Specifies the time (age points) at which the device degradation is calculated.

The default is `10`.

### GUI Equivalent

| | |
|---|---|
| Command | *Reliability Options – Basic – Aging Options – Single Age Point* |
| Field | *Age* |

## mosAgingTimeUnit

```
spectre.relxOpts mosAgingTimeUnit string {Years | Days | Hours | Minutes | Seconds}
```

### Description

Specifies the unit of the age point set by the `mosAgingTime` environment variable. Possible values are `Years`, `Days`, `Hours`, `Minutes` and `Seconds`.

The default is `Years`.

### GUI Equivalent

| | |
|---|---|
| Command | *Reliability Options – Basic – Aging Options – Single Age Point* |
| Field | *Age Unit* |

## mosCalcLifetime

```
spectre.relxOpts mosCalcLifetime boolean { t | nil }
```

### Description

Specifies whether to calculate the lifetime of the devices. Lifetime is calculated only for the devices whose electrical properties have degraded by at least the value set by the `mosDeltad` environment variable.


© 2023 Cadence Design Systems, Inc.

The default is `nil`.

### GUI Equivalent

| | |
|---|---|
| Command | *Reliability Options – Basic – Lifetime Calculation* |
| Field | *Calculate Lifetime of Devices* |

## mosDeltad

```
spectre.relxOpts mosDeltad string a_string_value
```

### Description

Specifies the threshold degradation value of the electrical properties (`Idsat`, `Vtlin`, `Idlin`) to calculate the lifetime of devices.

The default is `0.1`.

For example, if you set the value of `mosDeltad` to `0.2`, lifetime is calculated only for the devices whose electrical properties have degraded by `20%` or more.

### GUI Equivalent

| | |
|---|---|
| Command | *Reliability Options – Basic – Lifetime Calculation* |
| Field | *Degradation Value* |

## mosHciAccuracy

```
spectre.relxOpts mosHciAccuracy string { liberal (1) | conservative (2) }
```

### Description

Specifies the method used to be used in the reliability analysis when performing integration and substrate current calculation. Possible values are:

- `liberal (1)`: Uses backward Euler integration. Sets $I_{sub} = 0$, when $V_{gs} < V_{th}$
- `conservative (2)`: Uses trapezoidal integration. Calculates $I_{sub}$, when $V_{gs} > V_{th}$.


© 2023 Cadence Design Systems, Inc.

The default is `liberal (1)`.

### GUI Equivalent

| | |
|---|---|
| Command | *Reliability Options – Modeling – Hot-Carrier Injection Modeling* |
| Field | *Accuracy* |

## mosHciIdMethod

```
spectre.relxOpts mosHciIdMethod string { ids | idrain | idstatic }
```

### Description

Specifies how the simulator obtains the drain current to perform reliability analysis. Possible values are:

- `ids`: Uses the static drain current, which is also called as the channel drain current.
- `idrain`: Uses the dynamic drain current, which is also called as the AC drain current.
- `idstatic`: Uses the terminal static current.

The default is `ids`.

### GUI Equivalent

| | |
|---|---|
| Command | *Reliability Options – Modeling – Hot–Carrier Injection Modeling* |
| Field | *Drain Current Calculation Method* |

## mosHciOpMethod

```
spectre.relxOpts mosHciOpMethod string { calculate | from simulator }
```

### Description

Specifies the method to be used for obtaining the substrate terminal current or the gate terminal current. Possible values are:


© 2023 Cadence Design Systems, Inc.

- `calculate`: Calculates the substrate or gate terminal current by using the model parameters.
- `from simulator`: Calculates the substrate or gate terminal current by using the simulation results.

The default is `calculate`.

### GUI Equivalent

| | |
|---|---|
| Command | *Reliability Options – Modeling – Hot-Carrier Injection Modeling* |
| Field | *Vth, Igate, and Isub Calculation Method* |

## mosMaskDev

```
spectre.relxOpts mosMaskDev boolean { t | nil }
```

### Description

Specifies whether to enable device masking. If enabled, you can include or exclude:

- models that belong to the subcircuits listed in the subckt list.
- devices that belong to the models listed in the model list.
- devices that are listed in the instance list.

Possible values are `t` and `nil`.

The default is `nil`.

### GUI Equivalent

| | |
|---|---|
| Command | *Reliability Options – Basic – Device Masking* |
| Field | *Mask Device* |


© 2023 Cadence Design Systems, Inc.

## mosMaskDevInclude

```
spectre.relxOpts mosMaskDevInclude string { include | exclude }
```

### Description

Specifies whether to include or exclude the following during reliability analysis:

- models that belong to the subcircuits listed in the subckt list
- devices that belong to the models listed in the model list
- devices that are listed in the instance list

Possible values are `include` and `exclude`.

The default is `include`.

### GUI Equivalent

| | |
|---|---|
| Command | *Reliability Options – Basic – Device Masking* |
| Field | *Include/Exclude Devices* |

## mosMaskDevInstance

```
spectre.relxOpts mosMaskDevInstance string any_string_value
```

### Description

Specifies the instances that you want to include or exclude when performing the reliability analysis.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Reliability Options – Basic – Device Masking* |
| Field | *Devices List* |


© 2023 Cadence Design Systems, Inc.

## mosMinAge

```
spectre.relxOpts mosMinAge string a_string_value
```

### Description

Specifies the smallest age value for which degraded SPICE model parameters are calculated. Use either decimal notation (`xx.xx`) or engineering notation (`x.xxe+xx`) to specify the value.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Reliability Options – Modeling – Age Modeling* |
| Field | *Minimum Age* |

## mosMode

```
spectre.relxOpts mosMode toggle (t t t) ( "hci" "nbti" "pbti" )
```

### Description

Specifies which of the following AGEMOS aging effect contributors are to be considered when calculating the device degradation results:

- Hot-Carrier Injection (HCI)
- Negative Bias Temperature Instability (NBTI)
- Positive Bias Temperature Instability (PBTI)

For example:

- To consider both HCI and NBTI effects, set the `mosMode` environment variable to:
  ```
  envSetVal("spectre.relxOpts" "mosMode" 'toggle '(t t nil))
  ```
- To consider both HCI and PBTI effects set the `mosMode` environment variable to:
  ```
  envSetVal("spectre.relxOpts" "mosMode" 'toggle '(t nil t))
  ```
- To consider both NBTI and PBTI effects, set the `mosMode` environment variable to:
  ```
  envSetVal("spectre.relxOpts" "mosMode" 'toggle '(nil t t))
  ```


© 2023 Cadence Design Systems, Inc.

- To consider all HCI, NBTI, and PBTI effects, set the `mosMode` environment variable to:
  `envSetVal("spectre.relxOpts" "mosMode" 'toggle '(t t t))`
- To consider only HCI effect, set the `mosMode` environment variable to:
  `envSetVal("spectre.relxOpts" "mosMode" 'toggle '(t nil nil))`
- To consider only NBTI effect, set the `mosMode` environment variable to:
  `envSetVal("spectre.relxOpts" "mosMode" 'toggle '(nil t nil))`
- To consider only PBTI effect, set the `mosMode` environment variable to:
  `envSetVal("spectre.relxOpts" "mosMode" 'toggle '(nil nil t))`

### GUI Equivalent

| | |
|---|---|
| Command | *Reliability Options– Modeling – Age Modeling* |
| Field | *Aging Effect Contributors* |

## mosOutputDevDegInFilePath

`spectre.relxOpts mosOutputDevDegInFilePath string` *`any_string_value`*

### Description

Specifies the path to the setup file containing multiple `vdd` values for the models in use.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Reliability Options – Degradation – Device Degradation (File)* |
| Field | *Setup File* |

## otherRXOptions

`spectre.relxOpts otherRXOptions string` *`any_string_value`*

### Description


© 2023 Cadence Design Systems, Inc.

Specifies the additional arguments for the Reliability analysis.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Reliability Options– Basic – Other Options* |
| Field | *Additional Analysis Arguments* |

## otherODDOptions

`spectre.relxOpts otherODDOptions string` *`output_device_degradation_options`*

### Description

Specifies the additional arguments for output device degradation.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Reliability Options– Degradation – Device Degradation* |
| Field | *Additional Arguments* |

### Example

```
envGetVal("spectre.relxOpts" "otherODDOptions")
```

- For Spectre simulator:

```
envSetVal("spectre.relxOpts" "otherODDOptions" 'string "top_deg_num=1 op=[vth
idsat idlin gm gds]")
```

- For AMS simulator:

```
envSetVal("ams.relxOpts" "otherODDOptions" 'string "top_deg_num=1 op=[vth
idsat idlin gm gds]")
```


© 2023 Cadence Design Systems, Inc.

## relxTranStart

```
spectre.relxOpts relxTranStart string a_string_value
```

### Description

Specifies the start time of the transient time window for the stress test.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Reliability Options – Basic – Stress Test* |
| Field | *Start Time* |

## relxTranStop

```
spectre.relxOpts relxTranStop string a_string_value
```

### Description

Specifies the stop time of the transient time window for the stress test.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Reliability Options – Basic – Stress Test* |
| Field | *Stop Time* |

## reportModelParamChanged

```
spectre.relxOpts reportModelParamChanged boolean { t | nil }
```

### Description

Specifies whether to generate the model parameter changes report (`.bm#`) file. Possible values are `t` and `nil`.


© 2023 Cadence Design Systems, Inc.

The default is `nil`.

### GUI Equivalent

| | |
|---|---|
| Command | *Reliability Options – Output – Reports* |
| Field | *Model Changes Report* |

## uriDebugMode

```
spectre.relxOpts uriDebugMode string { 0 | 1 }
```

### Description

Specifies whether to print the debug information. Possible values are:

`0`: Does not print the debug information.

`1`: Prints the debug information.

The default is `0`.

### GUI Equivalent

| | |
|---|---|
| Command | *Reliability Options – Modeling – Unified Reliability Interface (URI) Models* |
| Field | *URI Debug Messaging* |

## uriLibs

```
spectre.relxOpts uriLibs string a_string_value
```

### Description

Specifies the path of the URI libraries.


© 2023 Cadence Design Systems, Inc.

The default is `""`.

### GUI Equivalent

| | |
|---|---|
| Command | *Reliability Options – Modeling – Unified Reliability Interface (URI) Models* |
| Field | *URI Libraries* |

## uriMode

```
spectre.relxOpts uriMode string { appendage | agemos }
```

### Description

Specify the calculation methods to be used in aging simulation. The possible values are `appendage` and `agemos`.

The default is `agemos`.

### GUI Equivalent

| | |
|---|---|
| Command | *Reliability Options – Modeling – Unified Reliability Interface (URI) Models – Age Calculation Approach* |
| Field | ■ `appendage`*: Append Age Method*<br>■ `agemos`: *Scale Param Method* |


© 2023 Cadence Design Systems, Inc.

# Environment Variables for Reliability Analysis

The following table maps the `spectre.relxOpts` environment variables with the corresponding reliability analysis options and parameters.

| Environment Variable | GUI Equivalent | Reliability Analysis Parameter |
|---|---|---|
| mosAgingTime | Age | `age time=[...]` |
| mosDeltad | Degradation Value | `deltad value=0.0` |
| mosMaskDevInclude | Include/Exclude Devices | `maskdev type=include` |
| mosMaskDevInstance | Devices List | `maskdev dev=[...]` |
| mosHciAccuracy | Accuracy | `accuracy level=1` |
| mosHcIdMethod | Drain Current Calculation Method | `idmethod type=ids` |
| uriLibs | URI Libraries | `urilib file` |
| uriMode | The Append Age Method and the Scale Param Method options | `urilib uri_mode=scaleparam` |
| uriDebugMode | URI Debug Messaging | `urilib debug=0` |
| relxTranStart | Start Time | `relxtran start=0.0` |
| relxTranStop | Stop Time | `relxtran stop=0.0` |
| degSortType | The Degradation Threshold and the Max Devices options | `degsort threshold=0.0` |
| degSortValue | The text fields next to the *Degradation Threshold* and the *Max Devices* options | `degsort number=0` |


© 2023 Cadence Design Systems, Inc.

| | | |
|---|---|---|
| gaAgeStepType | Type | `gradual_aging_agestep type=lin` |
| gaAgeStepStart | Start Time | `gradual_aging_agestep start=0` |
| gaAgeStepStop | Stop Time | `gradual_aging_agestep stop=0` |
| gaAgeStepTotalStep | Total Steps | `gradual_aging_agestep total step=1` |
| gaAgePoints | Age | `gradual_aging_agepoint points=[...]` |
| checknegagingtype | Negative Aging Check Level | `check_neg_aging type=error` |
| checknegagingclamp | Clamp Degradation Values for Negative Aging | `check_neg_aging clamp=no` |

To locate the GUI options corresponding to the environment variables, refer to the *GUI Equivalent* section of the variables.


© 2023 Cadence Design Systems, Inc.

© 2023 Cadence Design Systems, Inc.

E

# Migrating From ADE L to ADE Explorer

The Virtuoso ADE Explorer provides a seamless migration from ADE L to ADE Explorer. The following sections provide details about how to migrate states or cellviews saved using ADE L into ADE Explorer:

- Migrating an ADE L Setup to ADE Explorer
- Customizing Menus in ADE Explorer

Video

For the video demonstration, see Migrating an ADE L Setup to ADE Explorer on Cadence Online Support.


© 2023 Cadence Design Systems, Inc.

# Migrating an ADE L Setup to ADE Explorer

You can use one of the following ways to load the design and setup saved using ADE L in ADE Explorer:

- Migrating ADE L Setup Using Load State Form
- Migrating an ADE L State by Opening the Cellview in ADE Explorer
- Migrating ADE L Setup Using SKILL

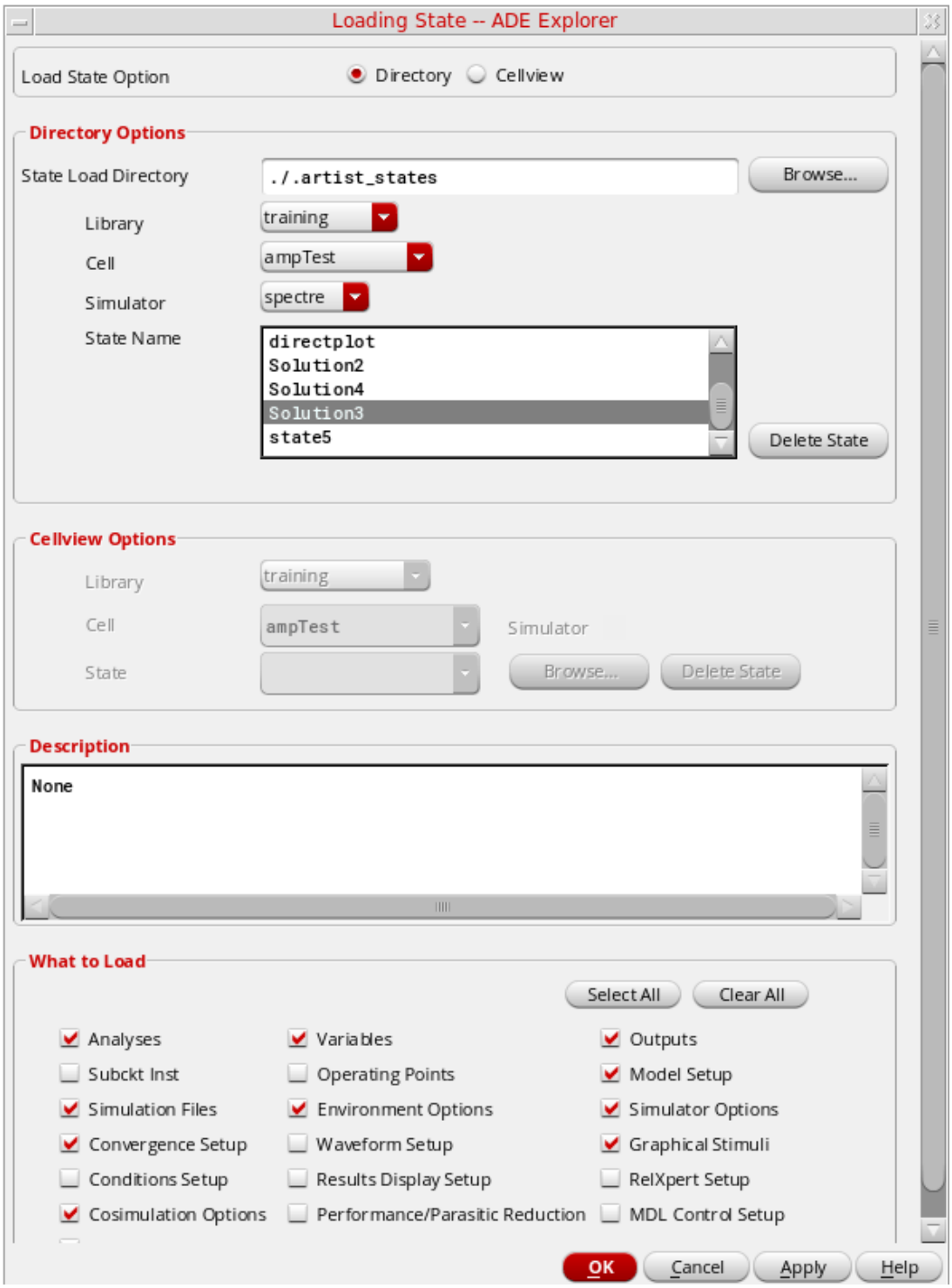
## Migrating ADE L Setup Using Load State Form

If you have a setup created using Virtuoso Analog Design Environment L (ADE L), you can open it with the ADE Explorer application. To do this, you need to have the setup states saved in the ADE L setup, which you can load in Explorer.

1. In CIW, select the design (library, cell, and view) that you want to load in Explorer and open the design in Schematic Editor. Now, follow the steps explained in the Opening Virtuoso ADE Explorer section to create a new cellview, `maestro`, and open ADE Explorer.


© 2023 Cadence Design Systems, Inc.

2. In ADE Explorer, choose *Session – Load ADE L State*. The *Loading State -- ADE Explorer* form appears.





© 2023 Cadence Design Systems, Inc.

In this form, select the previously-saved ADE L state and specify the following options:

3. Select one of the option buttons *Directory* or *Cellview* to indicate that you want to load the state from a directory or from a cellview. The default option is *Directory*.

   **a.** When the *Directory* option is selected, the fields in the *Directory Options* group box appear enabled. You need to specify a path in the *State Load Directory* by typing it in or by using the *Browse* button to locate it. Select the desired values in the *Library*, *Cell*, and *Simulator* fields. *State Name* specifies all the saved states for the cell and simulator combination that you specified. Select the state name you want to load.

   **b.** When you select the *Cellview* option, the fields in the *Cellview Options* group box appear enabled and those in the *Directory Options* group box appear disabled. You need to specify a *Library*, *Cell* and *State* you want to load. You may either type these in or specify them using the *Browse* button.

4. The *Description* field shows a short description about the selected state.

5. The substates displayed in the *What to Load* section are based on the substates in the selected state. You can enable or disable any of the substates individually by selecting the checkboxes next to them. You may also select all of them or none of them collectively by using the *Select All* or *Clear All* checkboxes at the upper left corner of the *What to Save* group box.

6. You can click *OK* to load the specified state.

   The system restores as much of the information as possible, ignoring settings from other simulators that are incompatible with the simulator that you have selected.

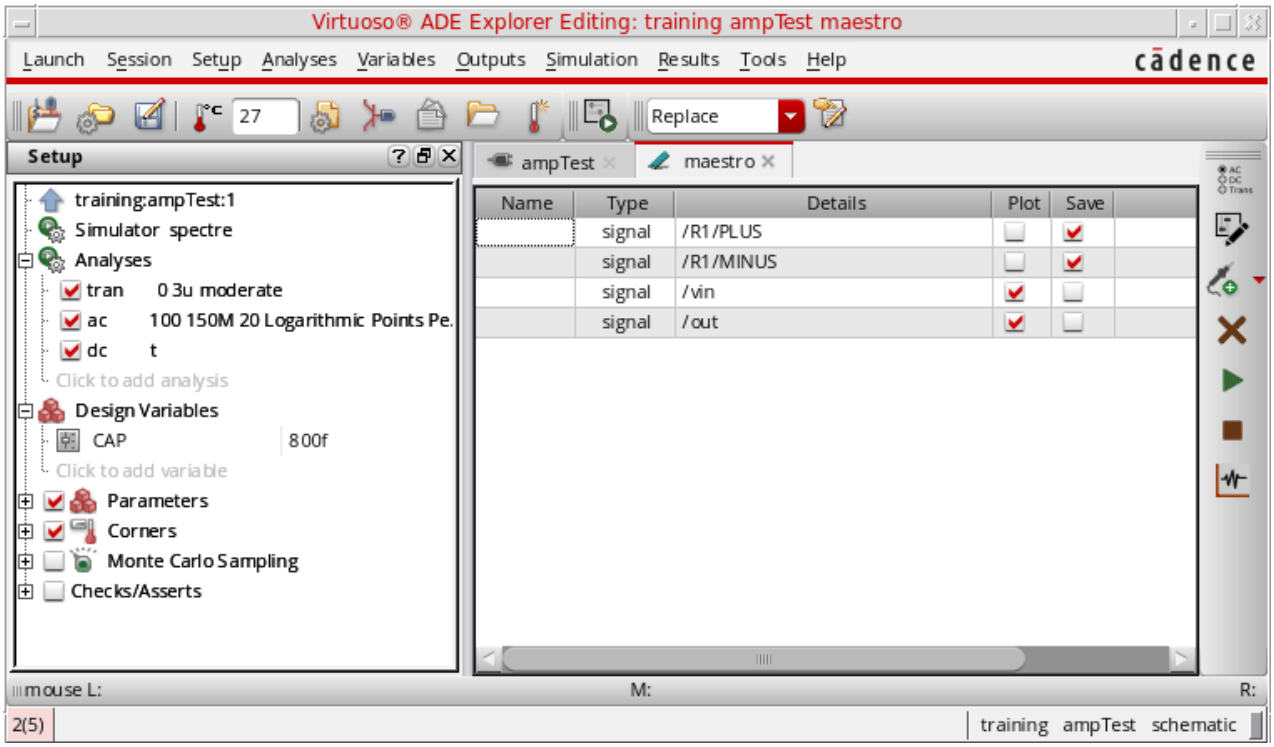
7. To delete a selected state, click the *Delete State* button.

**Note:** While loading a state from one simulator to other, the variable:

- may have the same name and value type. In this case, the variable is loaded.
- may not exist. In this case, no warning appears.


© 2023 Cadence Design Systems, Inc.

- is the same but the value type may be different or the value is not in the displayed list. In this case, an error is issued.




*Important*

Note the following important points:

- After creating a maestro view, any changes done to the ADE L Setup are saved in the maestro view and cannot be saved back to the original ADE L state.
- By default, the maestro state is saved in XML file format, which saves fewer files as compared to the number of files saved in an ADE state.
- When a Spectre simulation is running in interactive mode, do not restore a simulation setup in which Spectre mode or parasitic reduction is enabled. Restoring the simulation setup will result in the simulation run being terminated.
- If you have some results already saved in the ADE L state you are loading, they are automatically plotted in a new Virtuoso Visualization and Analysis XL graph window after the state is loaded in ADE Explorer.
- While loading an ADE L setup in ADE Explorer, the scalar values saved in the Output Setup of ADE L are displayed in the Value column of the Outputs Setup pane in ADE Explorer.


© 2023 Cadence Design Systems, Inc.

### Loading a State From Previous Tool Version

When you load a state saved from a previous tool version, the *Analog Design Environment State File Warning* form appears.

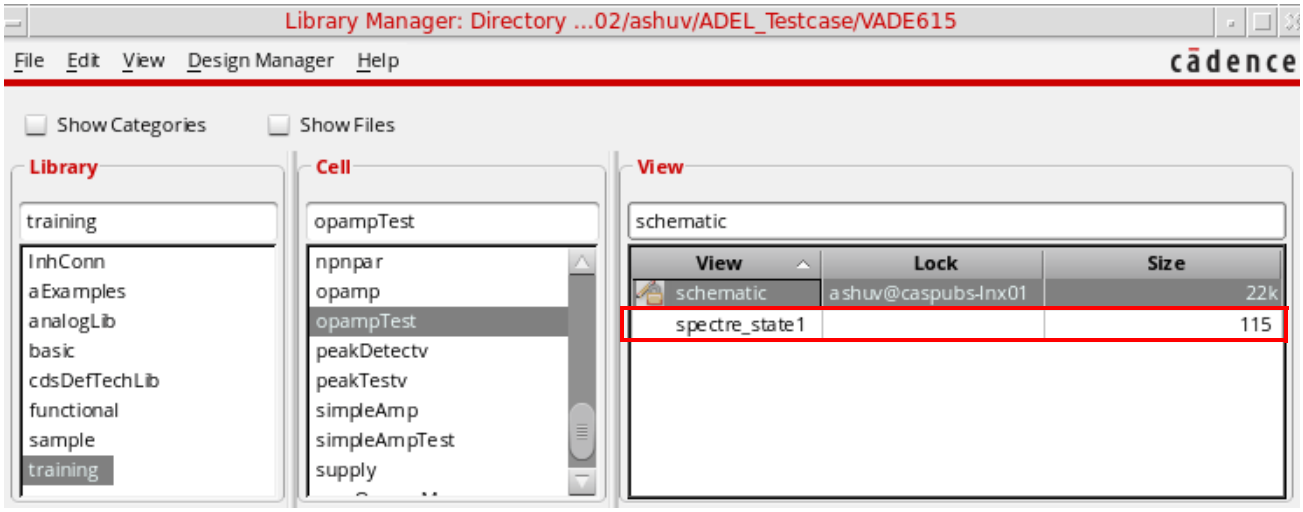
When you click *Yes*, the state file from the older tool version is converted into the current tool version and the converted file is loaded in ADE Explorer. When you click *No*, the form closes.

## Migrating an ADE L State by Opening the Cellview in ADE Explorer

Before opening an ADE L state in ADE Explorer, ensure that the setup state is saved as a cellview. To save cellview states, in the Loading State -- ADE L form, specify the *Load State Option* as `Cellview` . For more information about how to save states in ADE L, refer to the Virtuoso Analog Design Environment L User Guide.

To open the setup created using ADE L in ADE Explorer, do this:

1. Open the Library Manager window. The saved cellview state is displayed in the *View* column, as shown in the figure below. The default name of the saved cellview state is `simulator_stateName`.




2. Right-click the state name, `spectre_state1`, and choose *Open With*.


© 2023 Cadence Design Systems, Inc.

The Open File form appears.

**Note:** Alternatively, in the CIW, choose *File – Open* and in the Open File form, select the library, cell, and view to be opened.

3. From the *Open with* drop-down list, select *ADE Explorer*.

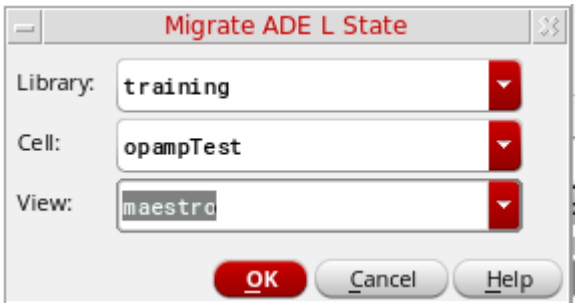
*Tip*

You can set the following environment variable to `ADE Explorer` to set this as the default value for the *Open with* drop-down list:

```
envSetVal("adexl.gui" "adestateDefaultApp" 'cyclic  "ADE Explorer")
```

4. Click *OK* to open the state.

The Migrate ADE L State form appears.

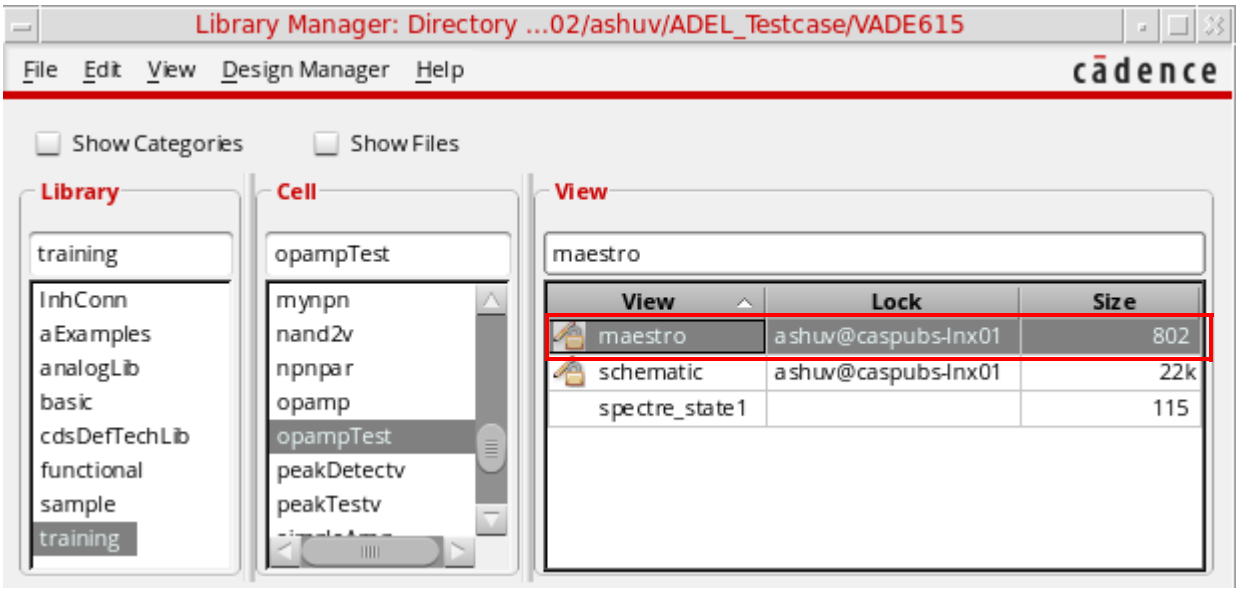



5. In this form, specify the view name with which you want to save the migrated view and click *OK*.

**Note:** The specified view is created and displayed in the *View* column of the Library


© 2023 Cadence Design Systems, Inc.

Manager window.




The Virtuoso ADE Explorer window appears automatically when this view is created.

## Migrating ADE L Setup Using SKILL

You can save an ADE L state as a cellview or a state file, and then migrate it to a maestro cellview using the `maeMigrateADELStateToMaestro` SKILL function.

The following example code migrates an ADE L state from a cellview state, `spectre_state1`:

```
maeMigrateADELStateToMaestro("Two_Stage_Opamp" "OpAmp_AC_top" "spectre_state1" )
```

This function creates a new cellview with the default name `maestro` and returns the names of library, cell, and view. For example, `Two_Stage_Opamp" "OpAmp_AC_top" "maestro"`

In the above example, the default name `maestro` was used to create a new cellview. This cellview will be available in the Library Manager and you can open it in ADE Explorer using the *Open With* command in the pop up menu.

The following example code migrates an ADE L state from a state file, `AC_state1`, saved in the `./libs/Two_Stage_Opamp/OpAmp/ade/test_states/` directory:

```
maeMigrateADELStateToMaestro("Two_Stage_Opamp" "OpAmp_AC_top" "AC_state1"
?maestroView "maestro1" ?migrateFrom 'directory ?statePath
"./libs/Two_Stage_Opamp/OpAmp/ade/test_states/")
```


© 2023 Cadence Design Systems, Inc.

This creates the following cellview: ("Two_Stage_Opamp" "OpAmp_AC_top" "maestro1")

The new cellview, `maestro1`, will be available in the Library Manager.


© 2023 Cadence Design Systems, Inc.

# Highlights of ADE Explorer

After you have migrated the ADE L state to ADE Explorer, notice the following new and powerful features of ADE Explorer:

- Improved Graphical-Interface
  - The ADE Explorer includes a revamped and user-friendly graphical user interface that makes its features more usable and convenient to access.
  - New menu commands, dockable assistants, and toolbar buttons have been introduced. The design and setup information, such as design variables, parameters, analyses are displayed in a new Setup assistant to the left.
  - A new Output Setup pane has been added in which you can setup the outputs to be saved or plotted during the simulation run.
  - Results tab is added next to the Outputs Setup tab that displays the results of the multi-point simulation run.
  - Virtuoso Visualization and Analysis XL Graph now appears as a dockable assistant with the ADE Explorer window.
  - Explorer Run Summary assistant is added that displays the simulation run status and summary.

  For detailed information about the ADE Explorer Graphical User-Interface, see Chapter 1, “Getting Started with ADE Explorer”.

- Real Time Tuning

  A new and powerful Real Time Tuning assistant has been introduced that you can use to tune your design and view the updated plots in the graph assistant simultaneously. With the help of this assistant, you can change the design variables and parameters values using the slider bars and instant output plots are generated. You can keep on tuning the design until you get the desired output plots.

  For detailed information, see Chapter 10, “Real-Time Tuning in Simulations”.

- Corners

  ADE Explorer provides the support to create and evaluate 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.

  For detailed information, see Chapter 6, “Working with Corners”.

- Monte Carlo


© 2023 Cadence Design Systems, Inc.

ADE Explorer provides the support to run Monte Carlo simulations to estimate parametric yields and generate information about the performance characteristics of the circuits you design.

For detailed information, see Chapter 8, “Performing Monte Carlo Analysis”.

- Checks and Asserts

  ADE Explorer provides the support to define circuit or device checks that are performed using the simulation results

  For detailed information, see Chapter 13, “Performing Circuit and Device Checks”


© 2023 Cadence Design Systems, Inc.

# Customizing Menus in ADE Explorer

The ADE Explorer menus are same as that of ADE L. However, some menus in ADE Explorer have been rearranged, removed, and some new menus have also been added for the new and advanced features.

When you migrate from ADE L to ADE Explorer, the menus customized using the above listed options can be reused in ADE Explorer, as described in the following sections:

- Reusing Changes from the .menu File
- Customizing Menus Using a Trigger in .cdsinit
- New Menus

## Reusing Changes from the .menu File

ADE L and ADE Explorer share the same `.menu` file. Therefore, if you have added, deleted, or changed any menu item in the `.menu` file for ADE L, the changes are automatically reflected in the `maestro` view opened in ADE Explorer for the same project.

## Customizing Menus Using a Trigger in .cdsinit

If you registered a callback using a trigger in `.cdsinit` to customize ADE L menus, you can register the same trigger callback for the `explorer` application type.

For example, if you have defined a trigger callback, `CCSadelExampleMenuTrigger`, to create a custom menu for ADE L, you can register the same for ADE Explorer by using the correct application name, as shown below.

```
(deRegUserTriggers "explorer" nil 'CCSadelExampleMenuTrigger)
```

Suppose you want to migrate an ADE L setup that has the `CCSUserPostInstallTrigger` trigger defined in the `.cdsinit` file. You can add the following to include the same menu in ADE Explorer:

```
(axlSessionRegisterCreationCallback 'connect_handlers)

(procedure (connect_handlers session_name)
  (axlSessionConnect session_name "postInstall" 'CCSUserPostInstallTrigger)
)
```


© 2023 Cadence Design Systems, Inc.

## New Menus

ADE Explorer includes some new menus for the new and advanced features.

*Important*

These new menus cannot be customized.


© 2023 Cadence Design Systems, Inc.

© 2023 Cadence Design Systems, Inc.

F

# ADE Explorer Frequently Asked Questions

This document contains the frequently asked questions and answers related to ADE Explorer.

## Why does Zij show large values sometimes?

This usually happens when a port in a component is open.

For example, consider a component where port 2 is open and $Z_{11}$ shows a large value, such as $1 \times 10^{12}$

Because port 2 is open, $I_2 = 0$

As $I_2$ and $I_1$ are proportionally related, $I_1 = 0$

Now, $Z_1 = \frac{V_1}{I_1} = \frac{V_1}{0}$ , which can make $Z_1$ reach infinity.

In such cases, you can use the equation for $Z_M$ to plot input impedance:

$$Z_M = \frac{(1 + S_{ii})}{(1 - S_{ii})}R$$

Where `R` is the reference impedance of the port of interest, `i` is the port number, and $S_{ii}$ are the S-parameters.

## How to run more than 10 OCEAN sessions at once?

By default, you can run a maximum of 10 simultaneous OCEAN sessions. If you try running more, the following error is displayed in the CIW:

```
Failed to lock log file: <Logfilepath>/CDS.log.9
```


© 2023 Cadence Design Systems, Inc.

```
*WARNING* file <logfilepath>/CDS.log.9 File is already locked by some other
process.
```

This is because the `CDS_LOG_VERSION` environment variable is set to sequential, limiting the maximum simultaneous runs to 10. With the default setting, log files such as `CDS.log.1`, `CDS.log.2`, and so on are created for each OCEAN session. For more information, see the *Log File Environment Variables* section in the *Cadence Application Infrastructure User Guide*.

To run more that 10 OCEAN sessions simultaneously:

1. Specify the log file name with the OCEAN command.

   ```
   ocean -log logfileName file.ocn
   ```

2. Set the environment variable `CDS_LOG_VERSION` to `pid`.

   ```
   setenv CDS_LOG_VERSION pid
   ```

This will create log files with the process ID as the extension. Therefore, there is no restriction on the number of OCEAN sessions that can be run simultaneously.

## How to get the list of output signals and expressions set using SKILL?

For an ADE L session, you can get the list of the outputs by using the asiGetOutputList SKILL function as illustrated below:

```
s=asiGetCurrentSession()
=>stdobj@0x1d14060

outs=asiGetOutputList(s)
=>(sevOutputStruct@0x15e71b8 sevOutputStruct@0x15e7208 sevOutputStruct@0x153ef48)

outs~>signal
=>("/out" "/net15" nil)

outs~>expression
=>(nil nil ymax(VT("/net15")))
```

For ADE XL, there is no direct SKILL function to get the list of outputs. However, you can write SKILL code to get a list of outputs defined for each test present in the current ADE XL session and return the signals or expressions listed for all the tests. Currently, the `asiGetOutputList` SKILL function does not list test names of the corresponding outputs.

To get a list of outputs for ADE XL:

1. Load the following procedure in the CIW:

   ```
   procedure(CCSGetOutputList()
   let((finalList sessId axlCv tests testId testSes)
   sessId = axlGetWindowSession(hiGetCurrentWindow())
   axlCv = axlGetMainSetupDB(sessId)
   ```


© 2023 Cadence Design Systems, Inc.

```
    tests = axlGetTests(axlCv)
    foreach(test cadr(tests)
    testId = axlGetToolSession(sessId test)
    testSes = asiGetSession(testId)
    finalList = append(finalList asiGetOutputList(testSes))
    )
    finalList
    ) ;let
    ) ;procedure
```

2. Open the ADE XL view for the design you wish to find the output list and enter the following procedure in the CIW:

   ```
   outs= CCSGetOutputList()
   ```

   The following will return the output list structure:

   ```
   (sevOutputStruct@0x1a0e5098 sevOutputStruct@0x1a0e50a8
   sevOutputStruct@0x1b506868)
   outs~>signal
   =>("/out" "/net15" nil)
   outs~>expression
   =>(nil nil ymax(VT("/net15")))
   ```

## How to read and load design variables from a text file to ADE L window?

You can use the following procedure to load the design variables from a text file with `.il` extension, for example, *`loadVarFile.il`*:

```
procedure(CCSAddVarFmFile(fileName)
let((myPort myList myVar)
  myPort = infile(fileName)
  unless(myPort
     error("\n File does not exist or is not accessible. Specify a valid filename")
  ) ;unless
  while(gets(myVar myPort)
       myVar = parseString(myVar)
       myList = append(myList list(list(car(myVar) cadr(myVar))))
  ) ;while
  asiSetDesignVarList(asiGetCurrentSession() myList)
  close(myPort)
t
) ;let
) ;procedure
```

To read and load the design variables:


© 2023 Cadence Design Systems, Inc.

1. Load the above procedure from `.cdsinit` file or from the CIW using the following command:

   ```
   load "loadVarFile.il"
   ```

   Alternatively, you can directly put the procedure in `.cdsinit` file or CIW to load the design variables.

2. Create a file containing the design variable names and their values defined in the following format:

   ```
   CAP 100p
   Res 310k
   x 4.0
   z 4f
   ```

   **Note:** There should not be any comments in the design variable file. Also, there should be no space after the last line otherwise a warning message will be displayed in the CIW.

3. Load the design variable file from the CIW using the following command:

   ```
   CCSAddVarFmFile("./designVarList.txt")
   ```

   **Note:** Before executing the above command, ensure that the ADE session is open in the active window because the procedure uses the asiGetCurrentSession() function.

## Why does the RTT assistant not evaluate expressions if the output format is set to SST2?

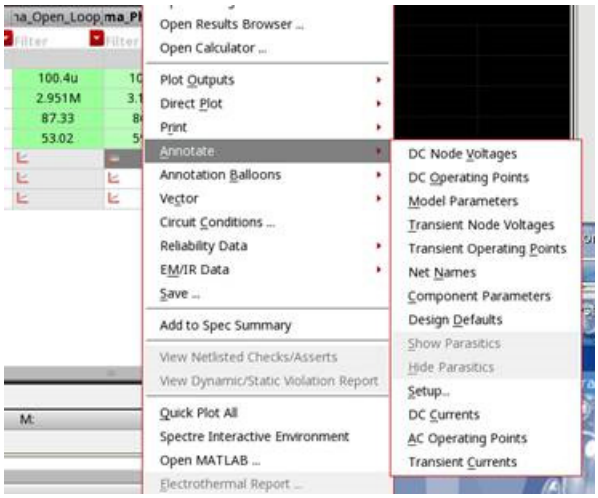
This is a known issue and currently no solution is provided for this.

To use the RTT assistant, you can set the output format to psf, psfxl, psf with floats or fsdb.


© 2023 Cadence Design Systems, Inc.

## Why do the schematic annotations not get updated automatically when a new simulation is run?

The solution is to annotate the operating points through the *Results* tab to refresh the directory path in the Annotation Setup form. To do this right-click a result name and select *Annotate*. You can then choose what to display.





© 2023 Cadence Design Systems, Inc.

© 2023 Cadence Design Systems, Inc.

# G

# Cadence© Load Balancing Software (LBS) Distribution Method

This chapter describes how to set up and use distributed processing for job simulation in Virtuoso using the Cadence Load Balancing Software (LBS).

See:


- Introduction to LBS Mode
- System Administration of Distributed Jobs with LBS
- Setting up DRMS in LBS Distribution Method
- Validating the LBS Distributed Processing Setup
- Submitting Simulation Jobs in LBS Distribution Method
- Monitoring Distributed Jobs in LBS Method
- Debugging an LBS Distributed Processing Setup
- Troubleshooting FAQs


## Introduction to LBS Mode

Distributed processing is the setup used for distribution of a program, task, or job across multiple hosts. In such a setup, the multiple hosts work together to complete the task or job. See System Administration of Distributed Jobs with LBS for more information.

By default, LBS uses the Cadence Queue Manager (`cdsqmgr`) to distribute jobs in the distribution queues. In addition, it provides a built-in hook into the following third-party Distributed Resource Management System (DRMS) tools:

- Load Sharing Facility (`LSF`) load balancer software from IBM.
- Sun Grid Engine (`SGE`) load balancer software from Oracle.


© 2023 Cadence Design Systems, Inc.

The LBS software consists of:

- An LBS client library: Used by Cadence applications to interact with the LBS server process.
- An LBS server program: Uses the LBS client library provided by the DRMS to interact with the DRMS cluster.

Therefore, the LBS server process is considered a DRMS client application.

# System Administration of Distributed Jobs with LBS

This section contains information that system administrators need to configure LBS and the distributed processing clusters. To use distributed processing successfully, ensure that your setup follows the requirements described in the following sub-sections:

- Common LBS Setup Requirements
  - Shell Variables for LBS Distribution Method
  - Additional Requirements

## Common LBS Setup Requirements

You must ensure that your site meets the following list of basic requirements or assumptions so that the distributed processing software works smoothly:

- To submit a job, you must have an account and a home directory on the submission host and the execution host.

  **Example:** If a user `morrison` who submits a job from `machine A` to `machine B` does not have an account on `machine B`, the submit request for `morrison` fails.

- You must have access to the OS specific Cadence® software on every execution host that might receive a job from you.
- The directory on your local machine from which you run the distributed processing software must be exported if you are using the `v/net`, `/home`, or `/hm`. Network mode also assumes that your working directory, and all your design data are under your account and are mirrored. Whenever the remote job references a path, that path goes directly back to the original directory on your home machine.
- The `uid` (UNIX user ID) of a user must not change when the user switches from one machine to another. For example, `uid` of `8819` must not identify user `morrison` on `machine A` and user `doe` on `machine B`.


© 2023 Cadence Design Systems, Inc.

- To submit jobs, all users must have write-access to the `/tmp` file system on the execution and submission hosts. The job submission and execution processes create certain temporary files that are saved in the `/tmp` file system.
- The setup considers that all files – input, output, error, and any command or OCEAN files – have absolute or relative locations, depending on whether they are preceded by a `/`. All relative locations are searched relative to the working directory of the job. Ensure that you specify the correct path for all files.
- The directory structure on the home machine and the remote machine must be identical. For example, if the working directory of the job on your machine is `/net/salmon/u1/morrison`, the design directory on the execution machine must be `/net/coho/u1/morrison`. If the directory structure does not match on `salmon` and `coho`, use a link to mirror the directory structure on the home machine.

### Shell Variables for LBS Distribution Method

The LBS distribution software supports the `cdsqmgr`, `LSF`, and `SGE` DRMS. To distribute and process jobs in the LBS method, you need to set the following environment variables based on the DRMS that you use for each distributed processing client machine:


© 2023 Cadence Design Systems, Inc.

`LBS_CLUSTER_MASTER` Set this to the LBS master host that controls job distribution to the queues. Your system administrator can help in this setup.

**Note:** You do not need to set `LBS_CLUSTER_MASTER` variable for LSF and SGE.

In .cshrc:

```
setenv LBS_CLUSTER_MASTER LBS_master_host
```

*`LBS_master_host`* is the host on which the LBS queue manager runs for controlling job distribution to the queues. Take your system administrator's help in setting this up. By default, LBS uses the `cdsqmgr` load balancer.

You can skip setting the `LBS_BASE_SYSTEM` variable if you use LBS.

`LBS_BASE_SYSTEM` Set this to define the load balancer that you want to use with LBS. For this, add one of the following commands in your `.cshrc` file.

- To use the default Cadence (`cdsqmgr`) load balancer

  ```
  setenv LBS_BASE_SYSTEM
  ```

  **Note:** You do not need to set the `LBS_BASE_SYSTEM` variable to use the default Cadence (`cdsqmgr`) load balancer. Optionally, if defined, it must be set to an empty value.

  See System Requirements for cdsqmgr Setup.

- To use the IBM Spectrum Load Sharing Facility (LSF) load balancer

  ```
  setenv LBS_BASE_SYSTEM LBS_LSF
  ```

  See System Requirements for LSF Setup.

- To use the Sun Grid Engine (SGE) load balancer

  ```
  setenv LBS_BASE_SYSTEM LBS_SGE
  ```

  See System Requirements for SGE Setup.

The variable values that you specify differ based on whether you use LBS with `cdsqmgr`, or LBS with third-party DRMS. However, ensure that you set the appropriate variables based on


© 2023 Cadence Design Systems, Inc.

the DRMS that you use. If you specify none of these variables, you can use distributed processing in the command-line mode only.

## Additional Requirements

In addition to the common and DRMS-specific requirements, ensure that your system administrator also considers the following:

- Time Clock Consideration
- Network Mode Consideration

### Time Clock Consideration

All users in a cluster must synchronize the clocks on all computers. Failure to synchronize the clocks can cause errors when you submit jobs. For example, machine Y's clock reports the time as 11:50, and machine Z's clock reports the time as 12:00. If a user on machine Y submits a job to machine Z to run 'Today' at 11:55, machine Z cannot run the job. The simulation will fail to run on machine Z because the specified run time occurred before the job was submitted.

### Network Mode Consideration

To run the distributed processing software, the default mode is the Network Mode. Network mode assumes that your home account is mirrored (linked) on every remote machine on which you might run a job, using commands such as, `/net`, `/home`, or `/hm`. Network mode also assumes that your working directory and all your design data are under your account and are mirrored. Whenever the remote job references a path, that path goes directly back to the original directory on your home machine.


© 2023 Cadence Design Systems, Inc.

# Setting up DRMS in LBS Distribution Method

This section describes the various DRMS-specific settings that are required for processing jobs using the LBS distribution method. You can set up your LBS setup depending on the DRMS that you choose.

- System Requirements for cdsqmgr Setup
  - Using cdsqmgr with remote shell (rsh)
  - Using cdsqmgr with secure shell (ssh)
- System Requirements for LSF Setup
  - Resource Requirement String Format
- System Requirements for SGE Setup

## System Requirements for cdsqmgr Setup

LBS `cdsqmgr` is a Cadence load balancing software that helps you set up a simple distributed system for multiple users based on the configuration file that you specify. It helps to run distributed processing jobs in queue mode.

Using LBS with `cdsqmgr`, you can set up any number of queues (collections of host machines) and hosts per queue. Each queue has a job-per-host limit. LBS `cdsqmgr` dispatches jobs to the hosts in the order that you specify.

Job distribution using LBS `cdsqmgr` is supported by the Cadence analog simulator Spectre, APS, XPS, and mixed-signal simulator AMS.

Setting up a site for an LBS system with `cdsqmgr` involves one of the following:

- Using cdsqmgr with remote shell (rsh)
- Using cdsqmgr with secure shell (ssh)

### Using cdsqmgr with remote shell (rsh)

To set up distributed processing with LBS `cdsqmgr`, ensure that you have the required system administrator rights or root permissions, and do the following:

1. Create the config file.
2. Check the host requirements. Ensure the following:


© 2023 Cadence Design Systems, Inc.

**a.** You must be able to log in to other hosts using `rsh` without a password.

**b.** All hosts must have a path set to the Cadence tools.

**c.** The `checkSysConf` command must run successfully on all hosts.

**d.** All hosts must be mapped to a common directory structure.

3. Start the daemon process.
4. Set up the UNIX environment for distributed processing users.
5. Run the configuration testing script.

**Note:** By default, the tool uses `rsh` to communicate between the hosts. Therefore, no additional settings are required to run `rsh`.

### *Creating the Config File*

Cluster configuration files can have any name and reside in any location. This file contains names of queues, machines in queues, number of hosts in a queue, names of hosts in a queue, and the job limit of individual hosts. The syntax for these files is as follows:

```
queue_1_name  N_-_number_of_hosts_in_queue_1
host_1        job_limit_for_host_1_on_queue_1
host_2        job_limit_for_host_2_on_queue_1
...
host_N        job_limit_for_host_N_on_queue_1

queue_2_name  N_-_number_of_hosts_in_queue_2
host_1        job_limit_for_host_1_on_queue_2
host_2        job_limit_for_host_2_on_queue_2
...
host_N        job_limit_for_host_N_on_queue_2
...
queue_n_name  N_-_number_of_hosts_in_queue_n
host_1        job_limit_for_host_1_on_queue_n
host_2        job_limit_for_host_2_on_queue_n
...
host_N        job_limit_for_host_N_on_queue_n
```

For example, the contents of a sample configuration file `lbs.config` are as follows:

```
************************ lbs.config ***************************
    queueA 2
    noi-user1 3
    noi-user2 2
    queueB 3
    noi-user1 4
    noi-user2 2
    noi-user3 3
    ************************************************************
```


© 2023 Cadence Design Systems, Inc.

Place this file in your `$HOME`.

In the above configuration file, `queueA` and `queueB` are the names of two queues. The first statement `queueA 2` indicates that the queue `queueA` has two hosts. These hosts are `noi-user1` and `noi-user2`. The statement `noi-user1 3` indicates that the host `noi-user1` can handle a maximum of `3` jobs. Similarly, host `noi-user2` can handle a maximum of `2` jobs.

### *Checking the Host Requirements*

1. For LBS `cdsqmgr` to work smoothly, you must be able to use `rsh` to log in to another host without a password. If you receive a prompt for a password, disable the password.
2. Ensure that all host machines can access the Cadence tools. To check if the path to the Cadence tool is available on each host, log in to all the hosts individually and run the following command in the Unix terminal window:

   ```
   > which spectre
   > which virtuoso
   ```
3. Use `rlogin` to log in to each host and check if the host passes the `checkSysConf` test. The following command ensures that the machine has all the required patches for running the tool.

   ```
   <ic_install_path>/tools/bin/checkSysConf <ic_install_version>
   ```

   For example, if you are running Virtuoso IC6.1.8, use the following command:

   ```
   checkSysConf IC6.1.8
   ```

   Repeat this test on each host for all Cadence tools (like simulators). For more details, refer to the Cadence article: Running checkSysConf to verify system patch levels.
4. Ensure that all hosts are connected to a common directory structure so that the project work area and the simulation directories are visible to each host.

### *Starting the Daemon*

*Tip*

A `daemon` is a program or process that runs as a background process.

To run `cdsqmgr` on the identified cluster master, do the following:

1. Identify the cluster master, which is the host on which the `daemon` needs to be run. If the master host is `noi-user1`, `rlogin` to host `noi-user1`.

   ```
   > rlogin noi-user1
   ```


© 2023 Cadence Design Systems, Inc.

2. Kill any `cdsqmgr` process that is already running under the root directory or any other account. Use the following command to find the process:

   ```
   Unix> ps -aef |grep cdsqmgr
   ```

   Running this command may result in an output as follows:

   ```
   userA 14494 1 0 19:05 ? 00:00:00 /ccstools/cdsind1/Software/IC618ISR_lnx86/
   tools/bin/32bit/cdsqmgr lbs.config
   ```

   Here, `cdsqmgr` is already running with `PID 14494`. To kill this `PID`, use the following command:

   ```
   > kill 14494
   ```

3. Kill any `cdsNameServer` process that is already running under the root directory or any other account. To do this, use the following commands:

   ```
   Unix> ps -aef |grep cdsNameServer
   userA 764 1 0 Jul30 ? 00:00:00 /ccstools/cdsind1/Software/IC618ISR_lnx86/
   tools/bin/32bit/cdsNameServer
   userA 4181 14330 0 20:47 pts/8 00:00:00 grep cdsNameServer
   ```

   Here, the `cdsNameServer` is already running with `PID 764`. To kill this `PID`, use the following command:

   ```
   Unix> kill 764
   ```

4. Start `cdsqmgr` as a root user on this master host (`noi-user1)` by providing the full path to the queue configuration file. The syntax is as follows:

   ```
   cdsqmgr <full_path_to_config_file>/lbs.config
   ```

   For example:

   ```
   Unix> cdsqmgr /home/user/lbs.config
   ```

**Note:** If a non-root user starts `cdsqmgr` from a user account on a cluster master machine, only that user will be able to submit jobs through that `cdsqmgr`. This is not recommended because jobs of individual users will not be queued together, and as a result, shared resources might not be used optimally.

### *UNIX Environment Setup for LBS Distribution Method*

To prepare your UNIX environment:

1. Ensure that your setup meets the common setup requirements. See Common LBS Setup Requirements.

2. Set the following in the `.cshrc` files for every user environment that intends to run the distributed simulation with LBS:

   ```
   setenv LBS_CLUSTER_MASTER LBS_master_host
   ```


© 2023 Cadence Design Systems, Inc.

Here, `LBS_master_host` is the host on which the `cdsqmgr` daemon is running with root permissions. See Starting the Daemon.

For example:

```
Unix> setenv LBS_CLUSTER_MASTER noi-user1
```

In the above command, `noi-user1` is set as the cluster master in every user environment.

### *Configuration Testing Script*

To test your setup for some common mistakes, run the `adptest` script. The script, together with the other Cadence binaries, is available at `<ic_install_path>/tools/dfII/bin`. To run the script, type the following at the terminal prompt:

```
adptest
```

The script displays messages in the same window where you run the script.

```
>>
BEGIN Analog Design Environment Distributed Processing Setup Test
Testing for existence of LBS_CLUSTER_MASTER variable...
    PASSED.
Testing opening of Analog Design Environment Job Server...
Launching Analog Design Environment job server...
Analog Design Environment job server has initialized!
    PASSED.
Please be patient - the remaining portion of the test takes some time.
Testing job submission to host `noi-user3' on queue 'queueB'...
    PASSED.
Testing job submission to host 'noi-user2' on queue 'queueB'...
    PASSED.
Testing job submission to host 'noi-user1' on queue 'queueB'...
    PASSED.
Testing job submission to host 'noi-user2' on queue 'queueA'...
    PASSED.
Testing job submission to host 'noi-user1' on queue 'queueA'...
    PASSED.
END Analog Design Environment Distributed Processing Setup Test
t
<<
```

### Using cdsqmgr with secure shell (ssh)

By default, Virtuoso uses `rsh` to utilize any distribution processing system. However, several organizations may block `rsh` and allow only the `ssh` setup, because of its higher security. You can configure Virtuoso to use `ssh` for distributed processing.

To configure Virtuoso for using `ssh`, ensure that you have the required root permissions or administrative rights, and do one of the following:


© 2023 Cadence Design Systems, Inc.

- Configure Remote Hosts to Accept ssh Connection without a Password
- Edit the cdsRemote.scr File
- Set Virtuoso Environment Variable in the .cdsinit File

### *Configuring Remote Hosts to Accept ssh Connection Without a Password*

For Virtuoso distributed system to work with `ssh`, you need to use `ssh` to log in to the execution hosts without any password. For example:

```
shyam@noi-userA$ ssh noi-userB
```

If you have `ssh` access on the execution hosts without any password, you immediately get a prompt on that machine as follows:

```
Last login: Mon Aug 19 17:23:58 2019 from noi-userA.cadence.com
----------------------------------------------------------------
This Cadence-owned computer system and Cadence's computer network are made
available to you for the purpose of conducting authorized company business.
You have no reasonable expectation of privacy with regard to content created,
stored and/or transmitted on this system, except as provided by applicable law.
Cadence may - at any time, for any reason and with or without notice - search,
monitor and inspect this computer and network to the full extent permitted by
applicable law.
================================================================
/usr/local/rc/cadence.login: No such file or directory.
[userA@noi-userB ~]$
```

If you do not configure `ssh` to work without password, the system may give you a message as shown below or may ask for the password.

```
The authenticity of host 'noi-userB (172.23.65.158)' can't be established.RSA key
fingerprint is ac:d9:69:47:10:8d:7f:83:86:53:5f:10:07:e3:20:82.
Are you sure you want to continue connecting (yes/no)?
```

You need to contact your IT department for creating a setup in which users can use `ssh` to log in to execution hosts without any password.

Alternatively, create an `ssh` setup that does not require a password. This method creates a private or public key and registers the execution hosts. Ensure that you have `ssh` installed on the machines. To create this setup, do the following:

1. Create a public or private key.
2. Copy the public key file as authorized_keys and add the remote host in the known_hosts list.
3. Locate the known_hosts files to validate addition of specified host.
4. Add more execution hosts in the known_hosts file.


© 2023 Cadence Design Systems, Inc.

### Creating a public private key

To create a public or private key, do the following:

1. Run the following Unix command:

```
unix> cd $HOME/.ssh
```

2. Run the following Unix command to create a public or private RSA key pair:

```
unix> ssh-keygen –t rsa
userA@noi- user A ~/.ssh]$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/userA/.ssh/id_rsa):
```

3. Press `Enter`. If an `id_rsa` file already exists, you get a prompt to overwrite the file (`Overwrite (y/n)?`). Type `y` and press `Enter` to overwrite the file.

4. Press `Enter` on the next prompt (`Enter passphrase (empty for no passphrase):`), to keep the passphrase null:

```
[userA@noi-userA ~/.ssh]$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/userA/. ssh/id_rsa)
Enter passphrase (empty for no passphrase): |
```

5. Press `Enter` on the next prompt (`Enter same passphrase again:`), press `Enter`.

```
(userA@noi- userA -/. ssh ]$ ssh- keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ userA /.ssh/id_rsa)
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
```

The following message appears:

```
[ user A @noi- userA ~/.ssh]$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ userA .ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ userA /. ssh/id_rsa.
Your public key has been saved in /home/ userA /.ssh/id_rsa.pub.
The key fingerprint is:
e2:5a:5d:d5:d6:cc:d7:d6:28:61:20:a4:51:27:71:63 userA @noi- userA
The key's randomart image is:
+-- [ RSA 2048]----+
|        .0*.E.0    |
|         0 * 0 0 =0|
|        .     0 + 0|
|             . 0 ..|
|        . S .      |
|       . 0 .       |
|        0 .        |
|       0           |
|      .            |
+-----------------+
[ userA @noi- userA ~/.ssh]$
```


© 2023 Cadence Design Systems, Inc.

The preceding procedure generates two files: `id_rsa` and `id_rsa.pub` in your `~/.ssh` directory.

### Adding remote hosts in the known_hosts list

To add the remote hosts in the `known_hosts` list, copy the `id_rsa.pub` public key file as `authorized_keys`. For this, run the following command:

```
unix> cat id_rsa.pub | ssh userA@noi-userB "mkdir -p ~/.ssh && chmod 700 ~/.ssh
&& cat >> ~/.ssh/authorized_keys"
```

You receive a prompt to confirm your connection to the host:

```
(userA@roi-userA ~/.ssh|$ cat id_rsa.pub | ssh userA@noi-userB "mkdir p -/.ssh &&
chmod 700 ~/.ssh && cat » ~/.ssh/authorized_keys'
The authenticity of host noi-userB (172.23.65.158) can't be established.
RSA key fingerprint is ac:d9:69:47:10:8d:7f:83:86:53:5f:10:37:e3:20:82.
Are you sure you want to continue connecting (yes/no)?
```

Type `yes` and press `Enter`. You are asked for the user password (`userA`).

```
[ userA@noi-userA ~/.ssh]$ cat id_rsa.pub | ssh userA@noi-userB "mkdir -p ~/.ssh
&& chmod
The authenticity of host 'noi-userB (172.23.65.158)' can't be established.
RSA key fingerprint is ac:d9:69:47:10:8d:7f:83:86:53:5f:10:07:e3:20:82.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'noi- userB,172.23.65.158' (RSA) to the list of known
hosts.
userA@noi- userB's password: |
```

After providing the password, the host `noi-userB` is added into the `known_hosts` file.

```
[ userA@noi-user A ~/.ssh]$ cat id_rsa.pub | ssh userA@noi-userB "mkdir -p ~/.ssh
&& chmod 700 ~/.ssh && cat » ~/.ssh/authorized_keys"
The authenticity of host 'noi- userB (172.23 65.158)' can't be established.
RSA key fingerprint is ac:d9:69:47:10:8d:7f:83:86:53:5f:10:07:e3: 20: 82.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'noi-userB, 172.23.63.158' (RSA) to the list of known
hosts.
userA@noi-userB's password:
[ userA@noi-userA ~/.ssh]$
```

Now, if you use `ssh` to connect to `noi-userB`, you will not get any query or prompt for entering a password.

### Validating the addition of new hosts

You will now find two more files, `authorized_keys` and `known_hosts` that are created in the `.ssh` directory. Run `ll` in the directory to find these files.

```
[userA@noi- user A ~/.ssh]$ ll
total 16
-rw-r--r-- 1 userA cadencel 399 Aug 21 15:36 authorized_keys
-rw------- 1 userA cadencel 1671 Aug 21 15:2G id_rsa
-rv-r--r-- 1 userA cadencel 399 Aug 21 15:20 ld_rsa.pub
```


© 2023 Cadence Design Systems, Inc.

```
-rw-r--r-- 1 userA cadencel 406 Aug 21 15:36 known_hosts
[userA@noi- userA ~/.ssh]$
```

Open the `known_hosts` file and validate that the host `noi-userB` is added.

### Adding more hosts in the known_hosts file

Repeat the command in Adding remote hosts in the known_hosts list to add more hosts (execution hosts) in the known_hosts file for the user. For example:

```
unix> cat id_rsa.pub | ssh userA@noi-user1 "mkdir -p ~/.ssh && chmod 700 ~/.ssh &&
cat >> ~/.ssh/authorized_keys"
```

Open the `known_host` file. You will find the `noi-userB` and `noi-user1` hosts added to it.

```
noi-userB, 172.23.65.158 ssh-rsa
AAAAB3NzaClyc2EAAAABIwAAAQEA5tUYB]hc3j+CLqydfvZ+DL5pUB9rqRfiXzS31lDyWFAN/
khLQSKy3Xg0pxbNRWZCcXnqnv2U0LGvTgIXe84spNsUYpYqoTn+3DlLn8MaqiEUK47oCB2WvW+tiDvLVM
ruCXy4XGi+hDwdtEEEgC++gwfJqONE5M6zRNh29pt65Qt3ZKzP3kPz340sqmTMkEluok8YMMa7TS+Iuxj
QphUGYTrTakUV3fqSSU+DZH43qdlplUUZKbPXgk3jK08suKsBdgXSpc9ey0oMswlmvu07THDqZexqMgSc
WqeERExH0850d5she3vvQ/BiJu525/Xa3U7sR3LvuK10nDWPCXtUQ==

noi- user1, 172.23.65.27 ssh-rsa
AAAAB3NzaClyc2EAAAABIwAAAQEA5tUYBjhc3}KLqydfYZ+DL5pUB9rqRfiXzS31lDyWFAN/
khLQSKy3Xg0pxbNRWZCcXnqYlV2U0LGvTgIXe84spNsUYpYqoTn+3OlLn8MaqiEtiK47oCB2WvW+tlOvL
VMniCXy4XGi<-hOwdtEEEgC++^/
3qONE5M6zRNh29pt65Ot3ZKzP3kPz340sqmTMkEluok8YMMa7TS+IuxjQphUGYTirakUV3fqS5U+DZfi4
3qdlplUUZKbPXgk3jK08suKsBdgXSpc9ey0oMswlmvu07THDqZexqMgScWqeERExH08 50d5she3vvQ/
BiJu525/Xa3U7sR3LvuK10nDWPCXtUQ==
```

### *Editing the cdsRemote.scr File*

By default, the `cdsRemote.scr` file, which is available in the `$CDSHOME/share/cdssetup/<username>/` directory, specifies the default configuration as follows:

```
remoteSystem=rsh // Default configuration
#remoteSystem=ssh
```

To use `ssh` instead of `rsh`, make the following changes in the `cdsRemote.scr` file:

```
#remoteSystem=rsh // To configure for generic ssh
remoteSystem=ssh
```

This enables the `remoteSystem=ssh` switch for the `ssh` method.

```
#----cdsRemote.scr = (/ccstools/cdsindl/Software/share/cdssetup/cdsRemsh)----#
------------------------------------------------------------------------------
exec "$remsh" "$rhostName" $noStdinFlag "$@"
}

#-----------------------------------------------------------------------
# Main program
#-----------------------------------------------------------------------

remoteSystem=rsh
```


© 2023 Cadence Design Systems, Inc.

```
#remoteSystem=ssh      # uncomment this line to configure for generic ssh

operation="$1"; shift
case "${remoteSystem}-${operation}" in
    rsh-shell)
         ClassicRsh "$@";;
    rsh-copy)
         exec rcp "$@";;
    ssh-shell)
         exec ssh "$@";;
    ssh-copy)
         exec scp "$@";;
    *)
```

### *Setting Virtuoso Environment Variable in the .cdsinit File*

To use `ssh`, set the following in your `.cdsinit` file:

```
envSetVal("asimenv.distributed" "remoteShell" 'string "ssh")
```

You can now run distributed LBS with `ssh` in Virtuoso.

Follow the steps in the "Using cdsqmgr with remote shell (rsh)" section to set up and configure LBS with `rsh`.

### Additional Requirements for LBS

- Check if `LBS_CLUSTER_MASTER` is set. If `LBS_CLUSTER_MASTER` is not set, then set this variable appropriately and proceed to the next step.
- Check if `LBS_BASE_SYSTEM` is set. If `LBS_BASE_SYSTEM` is set, ensure that it is set to `LBS_DEFAULT`.
- Check if `cdsqmgr` is running on the machine specified as `LBS_CLUSTER_MASTER`. If `cdsqmgr` is not running, kill `cdsNameServer` and re-run `cdsqmgr` from the root. See Starting the Daemon.
- Ensure that only one copy of `cdsqmgr` is running from the root account. If multiple copies of `cdsqmgr` are running, kill all the copies and `cdsNameServer`, and rerun `cdsqmgr` as root. See Starting the Daemon.
- Check if `cdsNameServer` is running on `LBS_CLUSTER_MASTER`. If it is not running, kill any `cdsqmgr` that may be running and rerun `cdsqmgr` as root. See Starting the Daemon.
- Ensure that you rerun `cdsqmgr` and `cdsNameServer` whenever the `queueConfig` file used by `cdsqmgr` is modified. See UNIX Environment Setup for LBS Distribution Method.


© 2023 Cadence Design Systems, Inc.

- Ensure that all machines specified in the `queueConfig` file are active and do not prompt you to enter the password at login. You can use `ping` to check if the machines are active. To check password-related issues, type `remsh <any_host_in_queue> pwd` in the submission host terminal.
- Take your system administrator's help to ensure that the timestamp of all the machines specified in `queueConfig` file matches with the machine specified as `LBS_CLUSTER_MASTER`.

## System Requirements for LSF Setup

In addition to the Common LBS Setup Requirements, ensure that the following initial setup requirements are met before you submit jobs to hosts that run the LBS distributed processing method with LSF. The LSF cluster must already be installed and configured as per the LSF installation guide. To troubleshoot any LSF installation issue, contact the LSF system administrator at your site.

Ensure that the LSF server version at your site is compatible with the LSF client version used by the Cadence LSF interface. If the LSF server and client versions (the version of LSF libraries integrated with Cadence LSF daemon) do not match and are not compatible, contact the LSF vendor (IBM) to determine the LSF version compatibility.

Compare the output of these commands:

- `$cdsfrb_lsf -version`

  The output determines the version of the LSF-provided client library used by the `cdsfrb_lsf` daemon.

- `$lsid –version`

  The output represents the minimum LSF cluster version for the integration.

You must set the environment variable `LBS_BASE_SYSTEM` to `LBS_LSF` in each shell from which you start a Cadence tool to submit jobs.

```
setenv LBS_BASE_SYSTEM LBS_LSF
```

Here, `LBS_LSF` specifies that you want to use the LSF load balancer.

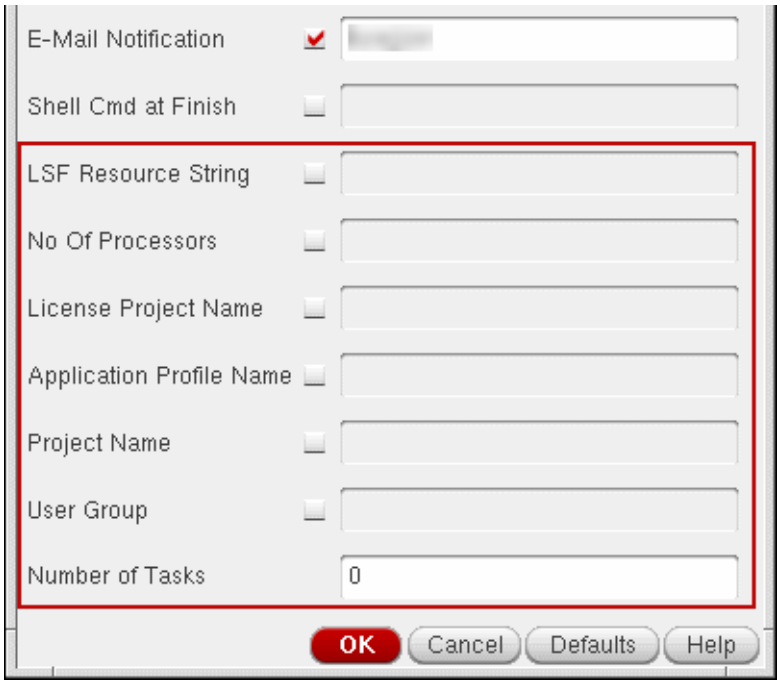
**Note:** You can skip setting the LBS_CLUSTER_MASTER variable for LSF and . Your local Cadence session will communicate directly with LSF.


© 2023 Cadence Design Systems, Inc.

### Additional Requirements for LSF

- Ensure that `LBS_CLUSTER_MASTER` is not set.
- Ensure that `cdsqmgr` is not running.
- Ensure that `LBS_BASE_SYSTEM` is set to `LBS_LSF`.
- Check if `LSF` software is installed and running on this machine. Ensure that `sbatchd`, `lim`, `pim` and `res` daemons are running on this machine, and that you can run the `lsload`, `bjobs` and `bsub pwd` commands without generating any errors.
- Ensure that machines specified in the LSF cluster are active and do not prompt you to enter a password when you log in to these machines. Use `ping` to check if the machines are active. To check for password-related issues, type `remsh <any_host_in_queue> pwd` in the submission host terminal.

In the LSF setup, the Job Policy Setup form displays a few additional fields as shown in the following figure:




These additional fields are:

- *LSF Resource String*: Lets you specify the additional resource requirements for the job that are not available as separate options in the Job Policy Setup form – for example, `mem` (available memory), `swp` (available swap space), `pg` (paging rate). You can specify the additional resource requirements using a valid LSF resource requirement string. For


© 2023 Cadence Design Systems, Inc.

more details and examples of LSF resource requirement string, refer to Resource Requirement String Format.

- *No of Processors*: Lets you specify the number of parallel processors that you must use to run the submitted job. To use the value specified in this field, select the check box next to this field.
- *License Project Name*: Lets you specify the name of the license project. To use the value specified in this field, select the check box next to this field.
- *Application Profile Name*: Lets you specify the name of an application profile that you must use to run the submitted job. To use the value specified in this field, select the check box next to this field.
- *Project Name*: Lets you specify the name of a project associated with the job. To use the value specified in this field, select the check box next to this field.
- *User Group* lets you assign the submitted jobs to the groups of user.

  **Note:** The group should not contain any sub-group.
- *Number of Tasks*: Lets you specify a numerical value that indicates the number of tasks in which TransNoise jobs needs to be divided for simulation. This option appears if you select the *Transient Noise* option for the *tran* analysis in the *Choosing Analyses* form. To achieve better performance with *Transient Noise*, specify the values of this field in a way that the number of runs divided by the number of tasks results in a whole integer value.

### Resource Requirement String Format

LSF uses the resource requirement string to select hosts for remote execution and job execution. For example, you can specify the following resource requirement string in the *<DRMS> Resource String* field, where *<DRMS>* stands for *LSF* :

```
select[swp > 50 && mem > 500]rusage[mem=100]
```

This resource requirement string runs a job only on a host that has 50 MB or more of available swap space and 500 MB or more of available memory, and also reserves 100 MB memory for the job.

The resource requirement string can have the following sections:

- Selection section (`select`) – Specifies the criteria for selecting hosts for a job.
- Ordering section (`order`) – Indicates how the hosts that meet the selection criteria must be sorted.


© 2023 Cadence Design Systems, Inc.

- Resource usage section (`rusage`) – Specifies the expected resource consumption of the job.
- Job spanning section (`span`) – Indicates if a parallel batch job must span across multiple hosts.
- Same resource section (`same`) – Indicates that all processes of a parallel job must run on the same type of host.

See Table G-1 on page 1673 for examples of each section.

**Note:** Ensure that you enclose arguments for each section within square brackets.

**Table G-1 Resource Requirement String Examples**

| Resource Requirement String | Description |
|---|---|
| `select[swp > 50 && mem > 500]` | Runs the job only on a host that has 50 MB or more of available swap space and 500 MB or more of available memory. |
| `order[cpu]` | Sorts hosts based on their CPU utilization and runs the job on the hosts with lower CPU utilization. |
| `rusage[mem=100]` | Reserves 100 MB memory for the job. |
| `span[hosts=1]` | Indicates that all the processors allocated to the job must be on the same host. |
| `span[ptile=2]` | Indicates that the job requests 2 processors on each available host, regardless of how many processors the host has. |
| `select[type===LINUX2.4 \|\| type==SOL10] same[type]` | Runs all parallel processes on the same host type – either LINUX2.4 or Solaris 10, but not both. |
| `select[linux &&mem>2000]` | Selects LINUX machines with memory greater than 2 GB. |
| `select[osVersion=='<version>' && mem>2000]` | Selects LINUX machines with the specified OS version and memory greater than 2 GB. |
| `select[mem>2000] order[-ncpus]` | Selects LINUX machines with a minimum number of CPUs and memory greater than 2 GB. |

For more information about resource requirement strings in LSF, see the product documentation for LSF at IBM Spectrum LSF.


© 2023 Cadence Design Systems, Inc.

Note the following:

- If the resource requirement string cannot be satisfied, the job will not run and the results are undefined.

  For example, if you specify `select[tmp>100]` as the resource requirement string and no machine in the queue has `100 MB` or more of memory in `/tmp`, the job will not run and no message will be displayed in the CIW window.

- If you specify a resource requirement string that conflicts with an option that you specify in the Job Policy Setup form, the job does not run and the results are undefined.

  For example, if you specify the host `tez` as the host to run the job under the *Hosts* section of the Job Policy Setup form and also select the `Only Use Selected Host` option and also specify the host name as `pvsol7` in the *LSF Resource String* field, the job will not be run and the status of the job will be undefined.

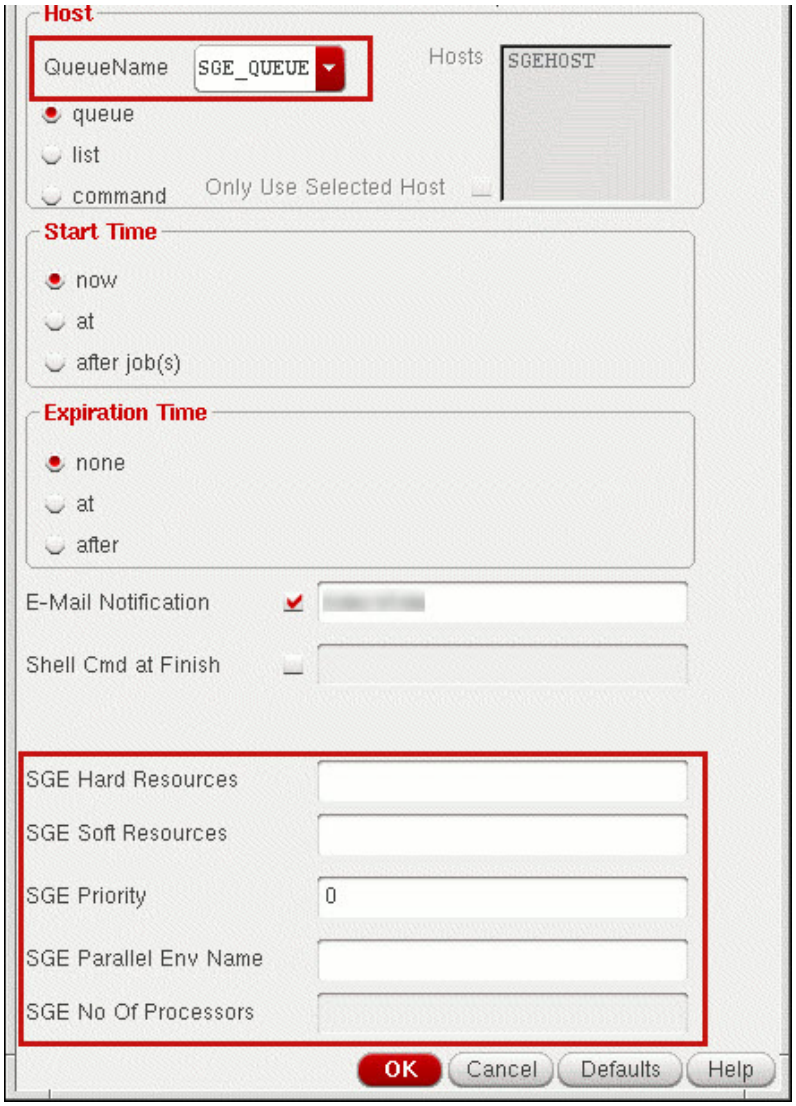
***Related Topic:***

- Simulator-Specific Environment Variables


© 2023 Cadence Design Systems, Inc.

## System Requirements for SGE Setup

In the SGE setup, the Job Policy Setup form displays a few additional fields as shown in the following figure:




These additional fields are:

- *SGE Soft Resources*: Specifies requirements for soft resources. You can specify values for `%MEM_EST` and `%CPU_EST` in this field.
- *SGE Hard Resources*: Specifies requirements for hard resources. You can specify values for `%MEM_EST` and `%CPU_EST` in this field.


© 2023 Cadence Design Systems, Inc.

- *SGE Priority*: Specifies the priority of the submitted job.
- *SGE Parallel Env Name*: Specifies the name of a parallel environment.
- *SGE No Of Processors*: Specifies the number of processors that you can use. Using this field, you can ensure that the job is run on a machine with sufficient number of CPUs available. You can also reserve these slots to stop many other jobs running on the same machine.

  **Note:** The *QueueName* field is also editable for SGE. Using this field, you can specify the name of a queue in which you want to run the job. The default value, `SGE_QUEUE` specifies that no queue name is provided by the user.

# Validating the LBS Distributed Processing Setup

To enable the distributed processing method, you need to specify a significant number of settings. Specifying the correct settings ensures that the setup is valid and appropriate messages are generated.

To specify the settings:

➥ Choose *Setup – Job Setup in ADE Explorer,* or *Options – Job Setup in ADE Assembler*.

The *Job Policy Setup* form appears.

The Job Policy Setup form tests the setup for enabling distributed processing and performs several checks to ensure that the distributed processing environment is set up correctly. The Job Policy Setup ensures the following:

- If the environment variables needed for distributed processing are set properly.
- If the LSF settings are correct, when you select LSF as the distribution method. It also verifies if you have set the path properly and the installed LSF version is supported.
- If the machines mentioned in the LBS queue are active and `cdsqmgr` DRMS is running on the cluster master machine. It also verifies the hierarchy setup for the machines in queue.

The Job Policy Setup performs all the specified checks and displays an appropriate `Error` or `Warning` message, if required. If an error occurs, it quits the subsequent checking, but continues to submit jobs with warnings. The Job Policy Setup uses `rsh` and `remsh` commands internally for performing the above mentioned checks. Therefore, you must ensure that these commands are run without a password. The Job Policy Setup performs only static checks. It does not submit test jobs to validate the setup.


© 2023 Cadence Design Systems, Inc.

**Note:** Ensure that there is no `echo` or `stty` command in your shell setup files because the check utility will not work properly if `echo` or `stty` commands exist in your shell setup files.

# Submitting Simulation Jobs in LBS Distribution Method

In distributed processing, the order of unused processing capacity of a job determines its assignment to a machine. The job-limit of the job has no impact on this assignment. Hence, when you use a DRMS such as LSF, distributed processing may send jobs to different machines in a different order, as compared to the jobs it sends without LSF.

In distributed processing, you can submit a job from one machine to another. These machines must have access to Cadence software, but they do not need to be in a predefined queue. For example, when you use the distributed processing mode with LSF, both the machines – the machine that you are submitting the job from and the machine running the job – must be in the cluster that has the license to use LSF.

When a job fails using the on of the supported DRMS, all its dependent jobs fail too. Jobs dependent on other jobs that fail are held in the DRMS database indefinitely until you use a DRMS command to force the job to run.

You can submit simulation jobs for distributed processing in one of the following ways:

- Using the ADE Assembler Graphical User Interface
- Using OCEAN
- Using Command-Line Mode

## Submitting Distributed Processing Jobs

To use distributed processing in ADE Explorer or ADE Assembler, you need the following:

- **Load Balancing Software (LBS)** to monitor the activity on the machines that you use for simulations. To know more about a default LBS setup, see System Requirements for cdsqmgr Setup.
- **Job Policy Setup form** lets you specify the time, the queue, or host for submitting your distributed job.
- **Job Monitor form** lets you view, suspend, resume, or kill jobs.

Additionally, command-line job submission and monitoring from the UNIX terminal window lets you submit, suspend, resume, monitor, or kill jobs.


© 2023 Cadence Design Systems, Inc.

You can submit simulation jobs using the Job Policy Setup form in ADE Assembler and ADE Explorer.

- To know more about submitting jobs in ADE Assembler, see Starting a Simulation.
- To know more about submitting jobs in ADE Explorer, see Starting a Simulation.

## Submitting Distributed Processing Jobs in OCEAN

This section describes how you can submit a distributed job using the Open Command Environment for Analysis (OCEAN). See the following sections:

- Preparing the OCEAN Environment for Distributed Processing
- Submitting a Job in OCEAN
- Selecting Results by Job Name
- Viewing Results

For more information, see the *OCEAN Reference*.

### Preparing the OCEAN Environment for Distributed Processing

This section describes the basics of setting up OCEAN to use distributed processing. For detailed information on distributed processing commands in OCEAN, see the OCEAN Reference manual.

To use OCEAN for distributing jobs, you must do the following:

1. Start OCEAN.
2. Enable distributed processing in OCEAN.
3. Disable distributed processing in OCEAN, when required.

#### *Starting OCEAN*

To start OCEAN, type the following at the command line of your terminal window:

```
>> ocean
```

#### *Enabling Distributed Processing in OCEAN*

To enable distributed processing, do one of the following:


© 2023 Cadence Design Systems, Inc.

- Type the following at the command line:

  ```
  >> hostMode( 'distributed )
  ```

- Set the `asimenv.startup hostMode` environment variable to `distributed` in the `.cdsenv` file.

### *Disabling Distributed Processing in OCEAN*

To disable distributed processing, type the following at the command line of your terminal window:

```
>> hostMode( 'local )
```

### Submitting a Job in OCEAN

To submit a job, use the `run` command, as follows:

```
>> ocean> run()
```

The job is submitted, and if the submission is successful, the `jobName` is returned. After you submit the job, use `simJobMonitor` to monitor the status of your job.

The terminal window running OCEAN may display the following message:

```
Job 'job004' ran on 'tez' and completed successfully
```

This message notifies you about the completion of a job. To turn off display of such messages, set the `asimenv.startup showMessages` environment variable to `nil` in your `.cdsenv` file.

**Note:** To stop or abort a running OCEAN script, use the `wait` command or the block argument with `run()`.

### Selecting Results by Job Name

To access results based on the job name, type the following on the OCEAN terminal:

```
>> openResults( jobName )
```

This is equivalent to calling `openResults` with a results directory path.

### Example

```
openResults( 'job001)
```

The preceeding command opens the results of `job001`.


© 2023 Cadence Design Systems, Inc.

**Note:** You can use the `openResults` command only after the job is complete.

### Viewing Results

Distributed processing in ADE Explorer or ADE Assembler creates a unique directory structure to contain your results. The results are stored in the following location on the submission host:

```
simulation/cell_name/simulator/schematic/distributed/job_name/psf
```

If you are using parametric analysis, the results are saved in the `/param` directory instead of the `/distributed` directory.

To know more about OCEAN commands for distributed processing, see the *OCEAN Reference*.

## Submitting Distributed Jobs in LBS using the Command Mode

The *command* option is meant for the advanced users who have full understanding of the DRMS (Distributed Resource Management System) commands and their corresponding syntax and options. Ensure that you enter a valid DRMS command to submit the job. The environment does not validate or check the command for errors because it assumes that any DRMS command that you enter is valid.

The command mode supports all DRMSs, especially LSF and SGE, that provide a command-line interface of the type <*job_submission_command job_command*>.

In general, the following assumptions are made about the command-line interface of the DRMS:

- The DRMS command-line interface accepts a single string as a job command. This string is the last argument of the job submission command.
- The DRMS job submission command is a non-blocking command. This means that the job submission command exits after queuing the job into the DRMS and does not wait for the job to complete.
- The DRMS job submission command returns the value `0` upon successful submission of the job. A non-zero return value indicates an error in job submission.

To use only the DRMS command mode for distributed processing, skip setting the value for `LBS_BASE_SYSTEM` and `LBS_CLUSTER_MASTER`.

**Note:** In this mode, the blocking mode and the autoplotting feature do not work, and all fields in the Job Policy Setup form (except *Job Name* and *Number of Tasks*) are disabled.


© 2023 Cadence Design Systems, Inc.

# Monitoring Distributed Jobs in LBS Method

The job monitor lets you view the status of submitted jobs. Using the Job Monitor, you can view the active jobs, save configurations, view log files, set job monitor options, set filters, or view job properties. For more information, see the *Cadence® Job Monitor Reference*.

# Debugging an LBS Distributed Processing Setup

You can debug your distributed processing setup based on the DRMS that you use.

See the following:

- Debugging LBS with cdsqmgr
- Debugging LBS with Third-Party DRMS

## Debugging LBS with cdsqmgr

In case of errors while running LBS with cdsqmgr, do the following:

1. Check that all the submission hosts and execution hosts pass the `checkSysConf` test. If `checkSysConf` fails, install the missing OS patches reported.

   You need to do this check for all IC tools and simulators that you run.

2. Check if the `/etc/hosts` file is set up properly. For example:

   a. Use:

   ```
   127.0.0.1 localhost.localdomain localhost
   ```

   b. Avoid:

   ```
   127.0.0.1 machine_name.machine_domain machine_name
   127.0.0.1 localhost.localdomain localhost
   machine_name.machine_domain machine_name
   127.0.0.1 localhost.localdomain localhost machine_name
   ```

3. Check if `cdsqmgr` has been started from the root. In the Unix terminal, use the `ps -eaf` command as follows:

   ```
   ps -eaf | grep cdsqmgr
   ```

4. Run the following command to check if `LBS_CLUSTER_MASTER` is set correctly. It must point to the host on which `cdsqmgr` is running.


© 2023 Cadence Design Systems, Inc.

```
echo $LBS_CLUSTER_MASTER
```

5. 5. Check that the `LBS_BASE_SYSTEM` variable is not set in your environment. On typing the following command, your system must return `Undefined variable`:

```
echo $LBS_BASE_SYSTEM
```

6. Ensure that the remote hosts can access the path to the design libraries, the project directory, and the path from where you launch Virtuoso.

7. Restart `cdsqmgr` if any information in the config file (`lbs.config`) contains updates, to make the changes effective for all users. For this, kill any running `cdsqmgr` and `cdsNameServer` process on `hostA` and start `cdsqmgr` again.

## Debugging LBS with Third-Party DRMS

You can debug the LBS settings based on the third-party DRMS that you choose for distributed processing of yours jobs. See


© 2023 Cadence Design Systems, Inc.

# Troubleshooting FAQs

This section describes how to troubleshoot some of the common difficulties that you can experience while running distributed simulations. See the following problem statements and their corresponding solutions:

1. Distributed processing does not work.
2. I specify an LSF Resource Requirement String in the Job Policy Setup form and submit a job. The job does not run even after a long time, and I do not see any error message.
3. I select LBS as the distribution method on the Job Policy Setup form, and the form expands, but I cannot use distributed processing.
4. I start the Job Monitor but it is blank. There are no jobs listed.
5. I name a job analysis2 and the system renames it to analysis2001.
6. The machine I want to use is not in the host list on the Job Policy Setup form.
7. When I click Run, I get a long error message that says that I cannot connect to the LBS queue manager.
8. When I run a distributed simulation, I get an error message saying that the job submission failed.
9. When I submit a job, it fails and comes back with an error message such as
10. I made job B dependent on job A. Job A ran, but job B did not. What is wrong?
11. My simulation finished, but I cannot find the PSF results.
12. I am using LSF with the distributed processing mode. When I try to run my jobs, I get an error message – “LSF setup is not proper”. I cannot submit any jobs.
13. The distributed processing mode randomly quits working. My jobs fail and I get an error message – “Could not connect to the LBS queue manager”. I usually keep a design framework II session running all night.
14. I am able to successfully submit distributed processing jobs to all hosts except the cluster master machine, even though it is in one of the cluster’s queues. The job status says: "Job dispatch failed; Cluster master timed out on ACK from execution host".
15. I encountered an error message containing a numeric identifier. For example:
16. What does this numeric identifier signify?
17. How do I specify a queue for a put together job?


© 2023 Cadence Design Systems, Inc.

18. LSF shutdown causes job distribution to freeze?

19. When I run a simulation successfully using LSF on Linux, and add new queues, the new queues do not reflect in the Job Policy Setup form.

**Distributed processing does not work.**

Use the following general troubleshooting tips as one of these could be the reason for the problem in a distributed processing run:

1. Ensure that when you use `rlogin` for logging in to remote hosts, you do not receive a password prompt. Authenticate the submission host as a trusted host by using `xhost`. Alternatively, create an `.rhosts` file in the home directory and ensure that it contains a '+' character.

2. Ensure that the remote hosts can access the paths to all the model files and the simulation directory.

3. Ensure that the remote hosts can access the Cadence tools hierarchy. To do this, use `rlogin` for logging in to the remote host. Run the following command and check if it returns the path to the Cadence hierarchy:

   ```
   unix> which virtuoso
   ```

4. Ensure that a user submitting a job has an account and a home directory on the submission host and the execution host.

   For example, if a user submits a job from machine A to machine B but does not have an account on machine B, the submit request fails.

5. Ensure that the directory on your local machine from which you run the distributed processing software is exported.

6. Ensure that the uid (UNIX user ID) of a user does not change as the user switches from one machine to another. For example, if the user `morrison` has uid `8819` on machine A, another user `doe` cannot use uid `8819` on machine B.

7. Ensure that all users who want to submit jobs have write-access to the `/tmp` file system on the execution and submission hosts. The host machines use this file system to save certain temporary files that are created as a result of the submission and execution processes.

8. Ensure that the directory structure on the home machine and the remote machine is identical. For example, if the job's working directory on your machine is `/net/salmon/u1/morrison`, the design directory on the execution machine must be `/net/coho/u1/morrison`. If the directory structure on `salmon` does not match that on `coho`, use a link to mirror the directory structure on the home machine.


© 2023 Cadence Design Systems, Inc.

9. Ensure that you specify the correct file paths. Distributed processing considers all files – input, output, error, and any command or OCEAN files – to have absolute or relative locations, depending on whether they are preceded by a forward slash (/). If relative, their location is searched relative to the working directory of the job.

**I specify an LSF Resource Requirement String in the Job Policy Setup form and submit a job. The job does not run even after a long time, and I do not see any error message.**

This can happen if you specify a resource requirement that is not possible to implement or conflicts with an existing option in the Job Policy Setup form. In such a case, run `xlsbatch` from the UNIX prompt to monitor the status of your job. `xlsbatch` is a job monitoring tool provided by IBM. To know more details, refer to the LSF documentation.

The status of a job may remain `PEND` for a long time because of a complication associated with the *LSF Resource Requirement String*. In the job details of `xlsbatch`, the `PENDING REASON` indicates why the job is in a `PEND` state. You can resolve this by modifying some of the job properties using *Job – Manipulate – Modify* in `xlsbatch`.

**I select LBS as the distribution method on the Job Policy Setup form, and the form expands, but I cannot use distributed processing.**

Distributed processing is available for the Spectre simulator and all the simulators integrated using OASIS. Ensure that you set the appropriate simulator in ADE Explorer or ADE Assembler. See the Command Interface Window (CIW) for error messages or warnings.

**I start the Job Monitor but it is blank. There are no jobs listed.**

This might be normal. The Job Monitor may not show any job for the following reasons:

- You have not started any jobs yet.
- You started jobs, but a long duration has elapsed and the jobs are finished and consequently terminated.
- The Job Monitor lists jobs by queue, usually in one queue at a time. If you are unsure about the queue that contains your jobs, select ALL under the Queues tab.

**I name a job analysis2 and the system renames it to analysis2001.**

This indicates that a job `analysis2` already exists in the same session. The distributed processing environment automatically appends a three-digit number to any duplicate job names to keep them unique.

**The machine I want to use is not in the host list on the Job Policy Setup form.**


© 2023 Cadence Design Systems, Inc.

Contact your system administrator, or cluster master operator, to add the machine to a queue. You can use the list field in the Job Policy Setup form to specify the machine you want to use.

**When I click Run, I get a long error message that says that I cannot connect to the LBS queue manager.**

Log on to the distributed processing cluster master and list the running processes. Find and kill the `cdsqmgr` process running under your user name. If you find this process, kill it.

Verify that a `cdsqmgr` is running under root. If it is not running, contact the cluster operator to restart it.

Exit the Cadence® software on your local host, and then restart it.

You can also get this error when you do not set your environment variables properly. Make sure that LBS_BASE_SYSTEM and LBS_CLUSTER_MASTER are set properly. Also verify that the master host is only in one cluster.

**When I run a distributed simulation, I get an error message saying that the job submission failed.**

Several situations can cause this error message:

- You do not have an account on the machine to which your job is submitted.
- The specified machine is disconnected from the network or shut down.
- The remote machine does not have access to the Cadence software, especially the distributed processing DRMS. By default, the local setting of the path environment variable should be valid for each execution host.

**When I submit a job, it fails and comes back with an error message such as**

```
sh: ./runSpectre not found, spectre not found.
```

The simulation fails because the remote job cannot find the simulator. You must have a path to the Cadence software on each of the remote machines. By default, distributed processing uses your local environment settings to initialize the session on the remote host. The search path on the remote host will be the same as the path setting on your local host. You must ensure that the path setting on the remote host is relevant, else the simulator will not be found.

For example, on your home machine you have a link to the Cadence software at `/u1/users/bill/Cadence`. However, your account is at `/usr1/mnt3/bill` on the machine that you run your job. The remote job goes to `/u1/users/bill/Cadence/tools/dfII/bin` on the remote machine and finds no account. You must either move the account to `/u1/users`, to match your home machine, or create a link from `/u1/users` to `/user1/mnt3`.


© 2023 Cadence Design Systems, Inc.

**I made job B dependent on job A. Job A ran, but job B did not. What is wrong?**

The dependency feature is not based on whether a job runs, but whether the simulation succeeds. If the simulation fails, subsequent dependent jobs will not run.

**My simulation finished, but I cannot find the PSF results.**

Distributed processing does not store its results in the standard `/psf` directory. Instead, it creates a `/distributed` directory at the same level as `/psf` and `/netlist`. The `/distributed` directory contains a directory for each job. This directory contains the `/netlist` and `/psf` directory for each job.

**I am using LSF with the distributed processing mode. When I try to run my jobs, I get an error message – "LSF setup is not proper". I cannot submit any jobs.**

When you use LSF, both the machines – the machine you are submitting the job from, and the machine you are submitting the job to – must be in the predefined LSF cluster. The scenario that you describe occurs if your home machine, or the machine that you are submitting from, is not in the LSF machine cluster.

**The distributed processing mode randomly quits working. My jobs fail and I get an error message – "Could not connect to the LBS queue manager". I usually keep a design framework II session running all night.**

The most likely reason is that your system administrator is making changes to the job queue. These changes may include changes to the depth or nature of the queue, or to the list of machines included in the queue. Each time the system administrator makes such changes this, they must shut down the Cadence Queue Manager (`cdsqmgr`). This disrupts any distributed jobs that you have in progress. You must exit your Cadence session and start again.

**I am able to successfully submit distributed processing jobs to all hosts except the cluster master machine, even though it is in one of the cluster's queues. The job status says: "Job dispatch failed; Cluster master timed out on ACK from execution host".**

Verify that when the `cdsqmgr` process starts on the cluster master, your PATH variable contains the path to the bin directory in the Cadence software hierarchy. This means that your `PATH` variable must contain `CDS_INST_DIR/tools/bin`).

**I encountered an error message containing a numeric identifier. For example:**

```
ERROR (28): There was an error connecting to the distributed job
server service. This host may not be set up correctly for using
Distributed Processing.
```


© 2023 Cadence Design Systems, Inc.

### What does this numeric identifier signify?

Each error message in distributed processing is given a unique numeric identifier. You can use this numeric identifier to access extended help (if available) on an error message by typing the following in the CIW:

```
msgHelp <prodID> <errorID>
```

where,

| | |
|---|---|
| `prodID` | is the product ID. In case of distributed processing, it is `'DP`. |
| `errorID` | is the error message ID number. This is the numeric identifier that is available in the error message box. |

For example,

```
msgHelp 'DP 28
```

This displays the extended help for the distributed processing error message, whose message ID is 28.

### How do I specify a queue for a put together job?

You can specify a queue for a put together job by specifying an environment variable as follows:

```
asimenv.distributed puttogetherqueue string "" nil
```

### LSF shutdown causes job distribution to freeze?

From the same terminal window that is running the workbench, kill the `cdsfrb_lsf` daemon and restart the workbench to get into distributed processing mode.

### When I run a simulation successfully using LSF on Linux, and add new queues, the new queues do not reflect in the Job Policy Setup form.

Ensure that you set the environment variable as, `LANG='C'`.


© 2023 Cadence Design Systems, Inc.